

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : **2000-082039**

(43)Date of publication of application : **21.03.2000**

---

(51)Int.Cl. **G06F 15/00**  
**G06F 12/00**  
**G06F 13/00**  
**G06F 17/30**

---

(21)Application number : **11-038422** (71)Applicant : **INTERNATL BUSINESS  
MACH CORP <IBM>**  
(22)Date of filing : **17.02.1999** (72)Inventor : **HIROSE SHINICHI  
KITAYAMA FUMIHIKO**

---

(30)Priority

Priority number : **10184392** Priority date : **30.06.1998** Priority country : **JP**

---

## (54) **DISPLAY CONTROL INFORMATION GENERATION METHOD AND COMPUTER**

(57)Abstract:

PROBLEM TO BE SOLVED: To produce a dynamic document or the display control information to provide a user interface according to each function to a client terminal having various functions by using a data object group that is independent of a terminal and the terminal attribute information to produce a view object group to produce the display control information respectively.

SOLUTION: A data object group that is independent of a terminal is produced in response to the display request given from the terminal. Then a view object group is produced by means of a data object group and the terminal attribute information and the display control information is produced by means of the view object group and by referring to the terminal attribute information. An HTML(hypertext markup language) generation mechanism 230 of an HTTP server 100 includes a part which produces a view object group by means of the data object group and the information received from a terminal DB 250 and the view object group itself.

---

## **CLAIMS**

---

[Claim(s)]

[Claim 1] A step which is the method of generating display control information as displayed with a form which suited a terminal which a user uses and receives a display

requirement from said terminalA step which answers said display requirement and generates a data object group independent of said terminal about display informationA step which generates a view object group for generating said display control information using said data object group and terminal attribute informationA display-control-information generation method which contains a step which generates said display control information by said view object group with reference to said terminal attribute information.

[Claim 2]The display-control-information generation method according to claim 1 which contains further a step which receives terminal identification information from said terminaland a step which acquires said terminal attribute information from said terminal identification information.

[Claim 3]The display-control-information generation method according to claim 1 with which a step which generates said view object group contains a step which generates a route object of said view object groupand a step which generates child objects by said route object.

[Claim 4]The display-control-information generation method holding directions information for which said data object is used in generation of a correspondence view objectand generation of said display control information by a view object according to claim 1.

[Claim 5]The display-control-information generation method according to claim 1 with which said terminal attribute information includes size information of a screen of said terminal.

[Claim 6]The display-control-information generation method according to claim 1 in which said display control information is a hypertext markup language (HTML).

[Claim 7]The display-control-information generation method according to claim 1 with which a step which generates said display control information contains a step which generates display control information of the portion in its duty with reference to said terminal attribute information sequentially from a route object of said view object group.

[Claim 8]A step which judges whether the display requirement concerned needs to generate said data object group by answering a display requirement from said terminalThe display-control-information generation method according to claim 1 which contains further a step to which a related view object in said view object group generates display control information corresponding to said display requirement when said data object group does not need to be generated.

[Claim 9]A step which is the method of controlling rerun / reconfirmation certificate of a page in a historyand carries out grouping of the page definitionA page definition which carried out grouping to directions information used for generation of said display control information A \*\*\*\* stepA step which distinguishes whether a rerun of a page in a history was performedA step which restores a processing state when it is rerun and cash of a processing state existsThe display-control-information generation method according to claim 1 containing a step which clears a processing state corresponding to execution of processing of applicationa step which stores a processing state at that time in cashand a page which specifies a group of said page definition and corresponds according to a result of processing of application.

[Claim 10]The display-control-information generation method according to claim 9 which contains further whether a page in a history is required for attestationa step which

checks an authentication state a step which checks attestation when attestation is required and a step which changes an authentication state of a page in a history by a result of processing of application.

[Claim 11] A computer which generates display control information so that it may display with a form which suited a terminal which a user uses comprising:

A module which answers a display requirement from said terminal and generates a data object group independent of said terminal about display information.

A module which generates a route view object for generating said display control information using said data object group and terminal attribute information.

A module which said route view object is ordered to generate the child view object.

A module which orders said route view object to perform generation of said display control information.

[Claim 12] The computer according to claim 11 which has further a module which derives said terminal attribute information from terminal identification information received from said terminal.

[Claim 13] The computer according to claim 11 by which said child view object can generate the child objects.

[Claim 14] The computer according to claim 11 by which said route view object can order said child view object to perform generation of said display control information.

[Claim 15] The computer according to claim 11 by which a view object group containing said route view object and said child view object generates said display control information with reference to said terminal attribute information.

[Claim 16] Answer a display requirement from said terminal and it is judged whether the display requirement concerned needs to generate said data object group The computer according to claim 11 which has a module which outputs said display requirement to said route view object when said data object group does not need to be generated.

[Claim 17] A module which is a computer which controls rerun / reconfirmation certificate of a page in a history and carries out grouping of the page definition A page definition which carried out grouping to directions information used for generation of said display control information A \*\*\*\* module A module which distinguishes whether a rerun of a page in a history was performed A module which restores a processing state when it is rerun and cash of a processing state exists The computer according to claim 11 which has a module which clears a processing state corresponding to execution of processing of application a module which stores a processing state at that time in a cache and a page which specifies a group of said page definition and corresponds according to a result of processing of application.

[Claim 18] The computer according to claim 17 which has further whether a page in a history is required for attestation a module which checks an authentication state a module which checks attestation when attestation is required and a module which changes an authentication state of a page in a history by a result of processing of application.

[Claim 19] As displayed with a form which suited a terminal which a user uses are the memorized storage a program which makes a computer generate display control information and said program A step which answers a display requirement from said terminal and generates a data object group independent of said terminal about display information to said computer A step which generates a route view object for generating

said display control information using said data object group and terminal attribute information. A storage which performs a step which said route view object is ordered to generate the child view object and a step which orders said route view object to perform generation of said display control information.

[Claim 20] Are a program which makes rerun / reconfirmation certificate of a page in a history control to a computer the memorized recording medium and said program. A page definition which carried out grouping to a step which carries out grouping of the page definition to said computer to directions information used for generation of said display control information. A \*\*\*\* step. A step which distinguishes whether a rerun of a page in a history was performed. A step which restores a processing state when it is rerun and cash of a processing state exists. The recording medium according to claim 19 which performs a step which clears a processing state corresponding to execution of processing of application. A step which stores a processing state at that time in cash and a page which specifies a group of said page definition and corresponds according to a result of processing of application.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention the control instruction for displaying a document etc. about the method and mechanism to generate with the form which suited the terminal which a user uses in more detail. Various terminals which a user uses for example PDA (Personal Digital Assistance) and the usual personal computer (PC). It is related with the method and mechanism which output HTML (Hyper Text Markup Language) etc. with the form which suited size transmission speed etc. of the display screens such as a cellular phone which has television or an information display function.

[0002] This invention relates to the mechanism which controls attestation and a rerun finely in consideration of the history function of a terminal in relation to the mechanism which generates a page dynamically [ the above ]. Even when performing in more detail the display which is a page which needs transactions and strict user authentications such as renewal of a database and a page is generated dynamically processing of control of a rerun a reconfirmation certificate etc. can specify easily from application and it is related with the mechanism in which it operates.

[0003]

[Description of the Prior Art] A what is called Singh (thin) client type system which the main part of application works by a Web server (application server) is increasingly used with osmosis of the technology of a Web base using a web browser in a user interface. At this time from a server the information sent to a client is form an included HTML document and in the terminal of a client. Application can be performed without working distribution installation etc. of application especially in advance of execution if even a web browser moves. Generally as compared with performing application by a client side a client side can be employed by hardware with lower throughput.

[0004] Conventionally low-end PC has been used for such a client terminal.

However PDA a hand-held PC still lower throughput and low cost [ in these days ] A set

top box and a new terminal equipment like the cellular phone which a web browser moves also by a car-navigation system and has a web browser function have also been developed. Therefore with PC which has a large viewing area for example the problem that user-friendliness will get very bad arose in PDA in which the Web page currently displayed well has a narrower viewing area such as coming to need frequent scroll operation.

[0005] As a method of coping with this problem as indicated to JPH10-133973A The resource of two or more HTML documents corresponding to the client terminal of all the target kinds and others which accompany them is prepared beforehand and there is a method of distinguishing the kind of terminal and returning a suitable thing for every request from a client. However it is necessary to create a lot of data in this case and there is a fault that the work load of systems development becomes huge. A work load when the kind of client terminal which should be supported increases is also large.

[0006] Only a PC-oriented standard HTML document is prepared as other methods and the system of changing the contents is also proposed according to the capability of a client terminal. This creates the reduced picture for example to a terminal with a narrow viewing area and sends to a client the HTML document changed so that the small picture might be referred to. The matter of transforming the HTML document prepared beforehand into JPH10-133973A described previously by a certain means so that two or more terminals can be suited is also indicated. However the concrete method about modification is not indicated.

[0007] JPH10-21165A has disclosed the communications system which edits the data size of the hypertext acquired from the server and data arrangement according to physical environments a user life scene etc. of a client which receive it and transmits. The performance of terminal such as communication line classification a workstation or a cellular phone a color number size of a picture etc. are included in these physical environments. A user life scene contains the data of a holiday etc. during attendance an opinion and enrollment. In this gazette using physical environments a user life scene etc. for choice of the information transmitted is indicated and it is not indicated in particular about screen constitution.

[0008] Furthermore T.W. Bickmore. To et al and "Digester: Device-Independent Access To The World Wide Web" (<http://www.fxpal.xerox.com/papers/bic97>). If proxy (proxy) receives the request of an HTML document a document is taken out from the specified HTTP (Hyper Text Transport Protocol) server Parsing that HTML document creating AST (Abstract Syntax Tree) and drawing up a new HTML document by performing predetermined modification operation to this AST is shown. In this Digester the HTTP server must draw up an HTML document first and also when making AST naturally that HTML document is needed. Since AST by which modification operation was carried out serves as an HTML document as it is the AST itself does not necessarily carry out control at the time of drawing up an HTML document.

[0009] Or about the rerun of a page. [ which can add the identification information and the state of a session to URL and HTTP header information (what is called Cookies) ]. (see function [ of Servlet ] (SUN Java Servlet API Specification Ver. 2.1) Session Tracking of IBM WebSphere etc.). However there are these as they carry out discernment and restoration of a state and control of the rerun of a page made into the purpose of this invention is not included.

[0010]About rerun control of the page in transaction processing of a database by which the technique of forbidding the rerun of the page which numbered URL and was displayed once is known widely. (For example it is introduced to 13.13 History Lists of RFC2068 HTTP1.1 Specification as "once-only" URL).

[0011]However programming will become complicated if it is going to change an authentication state and rerun control dynamically with processing of application or the diversity of a terminal (flexible processing is impossible and it becomes impossible for all the places that can be rerun to also rerun it originally only by a fundamental function). Since the page is not systematized specification of the page of such a processing program becomes complicated and if an addition deletion etc. which are pages start it will be necessary to rewrite the whole and the productivity and conservativeness and reliability of software will fall. It is difficult to control by a system which divides a page automatically with a terminal by such a system.

[0012]

[Problem to be solved by the invention] This invention is generating the dynamic document or display control information for providing the user interface according to each capability to the client terminal which has various capability.

[0013] It is also the purpose to provide the mechanism in which it is made not to depend for the application carried out by a server on the client terminal which uses the processing result of the application concerned.

[0014] Even if creation of the display document showing processing and the processing result concerned of the application carried out by a server is separated and various client terminals appear it is that what is necessary is to change only the creation portion of a display document.

[0015] An object of this invention is to provide the structure which controls the rerun and reconfirmation certificate further to operation and a demand of a up to [ the page of the past which used the history function on a terminal ] and the mechanism of specifying the control content simply from an application program.

[0016] This structure to control aims at making operation possible also when a page is dynamically generated by application in accordance with a terminal. The influence by dynamic generation and application of a page is eliminated from an application program and it aims at enabling it to specify the aforementioned control simply.

[0017]

[Means for solving problem] This invention is provided with the following.

The step which is how to generate display control information (for example HTML) as displayed with the form which suited the terminal which a user uses and receives a display requirement from a terminal.

The step which answers a display requirement and generates the data object group independent of a terminal about display information.

The step which generates the view object group for generating display control information using a data object group and terminal attribute information.

The step which generates display control information by a view object group with reference to terminal attribute information.

Thus if the portion and request content in which a data object group generates a data object group though a new terminal comes to output a display requirements since a terminal is generated independently are the same the data object group of change itself is

unnecessary. Terminals are apparatus such as usable TV and a cellular phone about the usual PC/PDA and a browser.

[0018] The step which receives terminal identification information from a terminal and the step which acquires terminal attribute information from terminal identification information can be included further. There are various methods in acquisition of terminal attribute information. Thus there are a method of making terminal identification information and terminal attribute information into a pair and memorizing them beforehand, the method of inputting when terminal attribute information is accessed first, etc.

[0019] The step which generates the view object group described previously can also be made for the step which generates the route object of a view object group and the step which generates child objects by a route object to be included. A route object may have a method which generates child objects with reference to terminal attribute information. These child objects generate a grandchild object if needed.

[0020] It can constitute so that the directions information for which a data object is used in generation of a correspondence view object and generation of the display control information by a view object may be held. With directions information, the priority of an object, division propriety information, outline propriety information, etc. are included. Terminal attribute information is the size information of the screen of a terminal and transmission speed, the aspect ratio of a screen or the information on a color number. Although display control information is a hypertext markup language (HTML) in the embodiment, other markup languages may be sufficient as it.

[0021] The step which generates the display control information described previously can also be made for the step which generates the display control information of the portion in its duty with reference to terminal attribute information to be included sequentially from the route object of a view object group. A route object generates the display control information about a whole page and child objects generate the display control information about the contents of the page. Since a view object refers to terminal attribute information, the display control information different [ but ] with this same view object may be generated.

[0022] In this invention, in being a display requirement from the same terminal and being a thing relevant to a former display requirement, the step which judges whether the display requirement concerned needs to generate a data object group by answering the display requirement from a terminal. When a data object group does not need to be generated, the related view object in a view object group may perform further the step which generates the display control information corresponding to a display requirement. If change is unnecessary to a data object, it can be coped with when a view object newly creates display control information. However, a data object group may not have to be created newly but \*\* may also have to generate a required view object.

[0023] As mentioned above, existing in a server before generating these since a data object and view object is also generated if needed. The module which answers the display requirement from a terminal and generates the data object group independent of a terminal about display information. The module which generates the route view object for generating display control information using a data object group and terminal attribute information. They are a module which a route view object is ordered to generate the child view object and a module which orders a route view object to perform generation of display control information.

[0024]The module which derives terminal attribute information from the terminal identification information received from the terminal may be included further.

[0025]The child view object described in the top has a method which generates the child objects and a route view object is able to constitute so that a child view object may be ordered to perform generation of display control information. The view object group containing a route view object and a child view object can also be constituted so that it may have a method which generates display control information with reference to terminal attribute information.

[0026]The program concerned may be stored in storage such as CD-ROM or FD when this invention explained above is mounted by a program.

[0027]In order to control a rerun a reconfirmation certificate etc. of the page in a history grouping of the page definition is carried out at the time of development of application. This group may also contain another subgroup and the whole page definition forms a tree structure. This page definition can be included in the directions information used for generation of said display control information. Grouping is performed to the page definition within this directions information. The set which is a page on which the set which is a page to which the rerun described later is forbidden and an authentication state are changed can be specified simply and flexibly only by this specifying the arbitrary group names of a page.

[0028]At the time of execution to the generation step of the aforementioned display control information in addition the step which distinguishes whether the rerun of the page in a history was performed The step which restores a processing state when it is rerun and the cash of a processing state exists The step which stores execution and the processing state at the time of processing of application in cash and the step which clears the processing state corresponding to the page which specifies the group of said page definition and corresponds according to the result of processing of application can be unified.

[0029]It adds to the step of the aforementioned display information generation and the step of rerun control of the page in a history When whether the page in a history being required for attestation and the step and attestation which check an authentication state are required the step which checks attestation and the step which changes the authentication state of the page in a history by the result of processing of application can be unified. However an application developer can direct change of an authentication state easily by specifying the group of a page definition like control of a rerun.

[0030]

[Mode for carrying out the invention] First the whole system about this invention is explained using drawing 1. The state where usual PC 120 the television 130 the portable computer 140 and the cellular phone 150 that builds in a web browser are connected with Web server 100 relevant to this invention in the network 110 is shown in drawing 1. PC 120 the television 130 and the portable computer 140 have a web browser and acquire information from HTTP server 100. Other apparatus which has a web browser may be sufficient as a client terminal. Since it has connected with the network 110 via radio the cellular phone 150 has been connected to the network 110 by the dotted line by drawing 1. Thus the computer linked to the network 110 differs in the size of a display screen transmission speed and throughput and operativity respectively. Therefore as for the document displayed on each display device it is preferred to suit the capability of each

terminal. Although the method of coping with it by the terminal side is also one of the methods of fitting the document to display to a terminal in this invention it has composition which explains HTTP server 100 below.

[0031] Then the functional block diagram of this HTTP server 100 is shown in drawing 2. The detailed block diagram of the sub functions of the control mechanism 210 in the functional block in HTTP server 100 is shown in drawing 3.

[0032] As shown in drawing 2 HTTP server 100 contains the control mechanism 210, DB240, the data service mechanism 220, the HTML mechanism of production 230 and terminal DB250. DB240 may not be included.

[0033] The control mechanism 210 receives the various parameters from the client terminals 120 thru/or 150 and replies the suitable HTML document corresponding to it. For this reason operation of the data service mechanism 220 and the HTML mechanism of production 230 is controlled. In Web application it may be one unit of processing with a series of requests in many cases from the same client terminal. Therefore even if the thing from [ out of two or more requests ] the same client is identified or generation of an HTML document is completed it is necessary to continue the execution as application by holding a state without the ability to open a resource wide immediately and preparing for the next access.

[0034] Here a series of accesses by a client terminal are called a session. For this reason it is judged whether the control mechanism 210 should manage a session and should start a new session for whether a certain request is a thing belonging to the existing session. In order to acquire terminal attribute information from the identification information obtained from the client terminals 120 thru/or 150 terminal DB250 is accessed and it may be made to output the terminal attribute information to the HTML mechanism of production 230. It may be made for each to access the data service mechanism 220 and the HTML mechanism of production 230 terminal DB250.

[0035] As shown in drawing 3 the control mechanism 210 comprises a function in which the session identification scheme 211, the page control mechanism 212, page control DB213, the application execution device 214 and the page rerun control device 215 were detailed in order to control a rerun, attestation etc. of a page.

[0036] It identifies which session the session identification scheme 211 analyzed the parameter sent from a terminal and has requested it from the terminal and data peculiar to a session is identified. a parameter is analyzed similarly it judges [ whether the page in the history of a browser is rerun and ] whether the page needs to be attested and the page control mechanism 212 a required page [ identify and ] Required information is taken out from page control DB213 and the processing state of the past required for attestation or a rerun is restored. By processing of the page control mechanism 212 finishing if such processing is unnecessary processing will move to the application execution device 214.

[0037] The application execution device 214 performs processing peculiar to Web application. For example in the case of banking application a actual banking transaction is performed. Based on this processing result the page rerun control device 215 updates the page control DB. That is when processing forbids the rerun which is the past page state cash required for the rerun of processing is removed from page control DB213.

[0038] The data service mechanism 220 is provided with the following.

The portion which accesses DB240 if needed and generates a data object group in order to carry out the processing demand of the retrieval required from the client terminals 120

thru/or 150etc.

The data object group itself which accesses to DB240and expresses the contentsand performs processing of a session.

This data object group is generated for every sessionand comprises this example as a tree structure which made the object showing the whole treatment process the route according to the document structure of a Web page. The object which hits a leaf corresponds to the data of a character stringa pictureetc. Howeverit is not limited to a tree structure.

[0039]The HTML mechanism of production 230 is provided with the following.

The portion which generates a view object group using a data object group and the information from terminal DB250.

The view object group itself.

A view object group will output HTML sequentially from a route view objectif a route view object receives the HTML generation instruction from the control mechanism 210.

The information acquired from terminal DB250 also on the occasion of this HTML output is used. The HTML document which the structure of the view object group generated from a data object group changes based on the information acquired from terminal DB250and is outputted by the view object group changes similarly. That iseven if the same data object group is generatedthe structure of a view object group may change and the HTML documents outputted even when a view object group is the same may differ. If the whole HTML document is generatedit will be transmitted to the client terminals 120 thru/or 150 of a requiring agency from the control mechanism 210. The portion and the view object group itself which generate a view object group can also refer to other parameters described later.

[0040]In present HTTP server 100generation of a dynamic HTML documentCGI.

(Common Gateway Interface) Or the structure

(SunJavaJDK1.2<http://jserv.javasoft.com/products/java-server/servlets/>) of a servlet. It is carrying out by using. Belowthe example realized using a servlet is described. This is because the continuous data-hold needed for Web application is easy and requirements required for this invention described below are satisfied. Howeverrealizing by CGI is also possible. In that casethe process started by CGI and the process of a background that interprocess communication is performed are used. It is also possible to realize a component required for this invention in the form of a Web substitute server.

[0041]The whole processing of HTTP server 100 is explained using drawing 4. Firsta display requirement is received from the client terminals 120 thru/or 150 (Step 1005).

The User-Agent field includedfor example in the header of URL (Uniform Resource Locator) and HTTP is used for this display requirement. For exampleURL is

"<http://aServer/servlet/RestaurantDemo>" and the User-Agent field of the header of HTTP serves as "deviceID". It uses for the User-Agent field acquiring the terminal attribute information used behind from terminal DB250. And it is judged from URL whether it is a new session (Step 1010). Since it is not put into session= <session number> in URL when it is a new sessionit shifts to Step 1020. On the other handsince it is put into session= <session number> in URL when it is not a new sessionthe session number is used.

[0042]In the case of a new sessiona data object group about display information as which the data access mechanism 220 was required first is generated (Step 1020). Generation processing of this data object is described later.

[0043]Nextthe HTML mechanism of production 230 generates a route view object (Step 1030). This view object has a method (or method requested from a child view object mechanism of production to generate a child view object) which creates a child view objectand a method which generates HTML. And a pair of a route view object and a session number is registered into a table. This table is used in Step 1010. And a view object group after a route view object is generated one by one using a data object group and terminal attribute information which were generated. About this processingan example is used and explained later. Generation of a required view object group will output a HTML generation instruction to a route view object from the control mechanism 210 (Step 1050). A route view object which received this generates taken charge HTMLand requests HTML generation from its child view object below. If it does soa view object below a child view object will generate its taken charge HTML (Step 1060). And the control mechanism 210 transmits a generated HTML document to a client terminal (Step 1065).

[0044]It is drawing 5 which showed the processing so far typically. Firstthe control mechanism 210 receives the User-Agent field and URL of HTTP. And the data service mechanism 220 generates a data object group from the information on URL received from the control mechanism 210. On the other handthe User-Agent field is used for acquiring terminal attribute information. And the HTML mechanism of production 230 is ordered for the control mechanism 210 to generate a route view objectand the HTML mechanism of production 230 generates a route view objectreferring to the route object and terminal attribute information of a data object group. The parent view object generates the child view objectreferring to a data object group and terminal attribute information also except a route view object. If a HTML generation instruction is taken out from the control mechanism 210 to the generated route view objectan HTML document as shown in drawing 5 will be generated by each view object after a route view objectand it will be outputted to a client terminal.

[0045]Since a session number will exist in URL when it is not a new sessionthe parameter received to the route view object corresponding to a session number is outputted (Step 1070). And it is judged from parameters other than the session number of URL whether a required view object is ending with generation (Step 1080). Even if the route view object itself makes this judgmentit may carry outwhen the view object below a route view object receives a parameter. If the required view object is generatedit will shift to Step 1060 (or step 1050). That iseach view object outputs HTML of the portion in its duty using terminal attribute informationand transmits the HTML document generated to the client terminal.

[0046]Step 1070 is further explained with reference to drawing 6. Step 1070 is further divided into a sub step for page controlsuch as attestation of a pageand rerun control of the past page. Firstby analyzing the parameter transmitted from the terminalit distinguishes whether control of a page is required (Step 1071)and when requiredpage control control is performed (1072). For examplesince it turns out that the past page was rerun by embedding the page identifier at URL and the HTTP header of the request when a user reruns the page in the history of a browserA processing state when the page is displayed on a key from the page control DB in the identifier is restored. To a page to be attestedit can attest by embedding the token of attestation at URL or a HTTP header similarly.

[0047] Step 1072 is further explained with reference to drawing 7. As page control control of Step 1072 as shown in drawing 7 specifically attestation of a page and rerun preparations of the past page are made. The above-mentioned figure expresses these sub steps. First it is distinguished whether the page in the history of a browser was used by analyzing the parameter transmitted from the terminal (Step 107201). Here since it is outside the processing object of this subflowchart when the new page which is not the past history is used it obtains and it escapes to the next processing as it is. (step 107202) which investigates whether the cash of the past processing state exists with reference to page control DB 213 when it is a page in the past history. Since I hear that the rerun is forbidden when cash does not exist it flies to applicable error-handling (step 107206). which restores a processing state using it when cash exists (step 107203). Next which attestation distinguishes using [ whether it is a required page and ] the tree structure of a page (step 107204). This distinction is performed by investigating whether that page belongs to the page group who needs attestation using the tree structure of a page. When attestation is unnecessary it escapes to the next processing of a subflowchart as it is. When attestation is required it confirms whether the invalidity of the attestation of the page is carried out by the authentication token contained in the parameter transmitted from the terminal or page control DB 213 and it is judged whether attestation is effective (step 107205). If attestation is effective it escapes to the next processing as it is. When invalid it flies to error handling of step 107206.

[0048] Again drawing 6 is referred to. Next original processing of application is performed (1073). For example if it is the application of banking a specified banking transaction is performed. At this time cash of the state after application execution is carried out to the page control DB in preparation for the rerun of a history page of a browser. It distinguishes whether page control was changed based on a processing result of this application (1074) and if required the page control DB is updated (1075). For example when a commitment of a database is performed it is necessary to forbid a rerun of a page of an update process of a database. That is processing state cash with which the page control DB corresponds is cleared. As another example when attestation is performed as a result of processing (or the attestation till then was canceled) renewal of an applicable portion of the page control DB is performed.

[0049] Step 1075 is further explained with reference to drawing 8. As renewal of page control DB of Step 1075 prohibition of cancellation of attestation of a page or a rerun of the past page is specifically performed. The above-mentioned figure expresses these sub steps. First it is distinguished whether a result of application execution (Step 1073) which is a step in front of Step 1075 is investigated and a certain page group's rerun is forbidden by the executed result (Step 107501). If it is necessary to forbid a rerun a page group will be computed based on a page group name specified with an application program and a tree structure of a page definition (Step 107502). Next which removes cash of a processing state corresponding to a page belonging to an applicable page group from page control DB 213 (Step 107503). If it is not necessary to forbid a rerun Step 107502 and 107503 will be skipped. By a processing result of application it is distinguished similarly whether attestation is cancelled or not (step 107504). If it is necessary to carry out whether it is invalid in attestation a page group which it is similarly made based on a tree structure of a page definition whether attestation is invalid will be computed (Step 107505). In step 107506 attestation of a page belonging to an applicable page group is

cancelled. If it is not necessary to cancel attestationStep 107505 and 107506 will be skipped.

[0050]Drawing 4 is referred to again. When it is judged that the required view object is not generated on the other hand it is judged whether a new data object is required (Step 1090). Since it may be unable to obtain the data in the data object generated before it when you need new data it requests to generate a data object in the data service mechanism 220 in being required (shift to Step 1020). Since the subtree of the original data object will be generated when a new data object is needed although it is the same session processing which registers the session number in Step 1030 is not performed. The view object corresponding to the newly generated data object is generated hereafter and a view object generates an HTML document.

[0051] When you do not need a new data object at Step 1090 a view object generates a required child view object with reference to a data object group and terminal attribute information and shifts to Step 1060 (or step 1050) (Step 1100). That is each view object outputs HTML of the portion in its duty using terminal attribute information and transmits the HTML document generated to the client terminal.

[0052] Then a part of processing after Step 1070 is outlined using the example of drawing 5. Drawing 9 shows the case where a user chooses ABC from the HTML document of drawing 5. deviceID shown in the useAgent field of HTTP is used like drawing 5. On the other hand URL serves as

"http://aServer/servlet/RestaurantDemo?session=1&ABC=selected" and it can recognize that it is the same session as drawing 5 by the portion of session=1. Although the portion of ABC=selected shows that ABC of drawing 5 was chosen it is judged in a view object that a new data object is required. If it does so the data service mechanism 220 will newly generate a data object group. And a view object group is generated using terminal attribute information. In this example when drawing up an HTML document with reference to terminal attribute information only the portion surrounded by the dotted line among view object groups outputs HTML. And the HTML document generated is expressed to the bottom. Here the NEXT button (or hyperlink) is generated.

[0053] A case where a user points to NEXT of drawing 9 is shown in drawing 10.

Handling of the useAgent field of HTTP is the same. URL serves as

"http://aServer/servlet/RestaurantDemo?session=1&next=pressed" and it can recognize that it is the session more nearly same than a portion of session=1 as drawing 5. A portion of next=pressed shows that the NEXT button of drawing 9 was pushed.

However it is judged [ that it can respond by the existing view object and ] in a view object. In this case it is not made but a dotted-line portion of a view object group outputs HTML and any change to a data object and a view object draws up a final HTML document. The PREV button (or hyperlink) and the NEXT button (or hyperlink) are contained in this HTML document.

[0054] In such a case the HTML example output and the example of operation of the NEXT button (or the PREV button for returning to the last page) which can be set are as follows.

[0055]

[Table 1]

<a href="/servlet/Restaurant?session=1&next=pressed"> Next </a> [0056] The contents of the anchor tag can also specify the image of not only a mere character string but a button.

URL with a parameter is described as an attribute value of an anchor tag. When a user clicks this hyperlink this URL is sent from a terminal a parameter is passed to the control device of this invention and a next=pressed parameter is passed to the view object group specified using the session parameter. It detects that the NEXT button was pushed by this next=pressed and the HTML generating program of a view object outputs the HTML command which is the next page.

[0057] The case where a user furthermore points to NEXT of drawing 10 is shown in drawing 11. The handling of the User-Agent field of HTTP is the same. URL serves as "http://aServer/servlet/RestaurantDemo?session=1&next=pressed" and it can recognize that it is the session more nearly same than the portion of session=1 as drawing 5. The portion of next=pressed and the internal state of a view object show that the NEXT button of drawing 10 was pushed. Here it is judged that a new view object is required for a view object group. In such a case a required view object is generated with reference to the data object group and terminal attribute information which were generated before. Here dotted-line portions other than a route view object are newly generated. And referring to terminal attribute information the view object group of the portion of a dotted line generates HTML and draws up an HTML document.

[0058] In a top the terminal attribute information which was not described in detail is described. Although it supposes that terminal attribute information will be acquired from deviceID contained in the User-Agent field of HTTP in the example of drawing 5 thru/or drawing 11 this matching information is included in terminal DB250. For example the correspondence table that the terminal attribute information is acquired from deviceID is prepared. To terminal attribute information a thing as shown in drawing 12 can be considered. For example ScreenSize (screen size) ScreenRatio (aspect ratio) ScreenQuality (definition of a screen) ColorDepth (color number) ModemSpeed (transmission speed) Image (is an image usable?) There are Table (is a table usable?) Frame (is a frame usable?) ScrollBar (can it scroll?) PointingDevice (is a pointing device usable?) etc. The example of each value is shown in the right column of drawing 12. All the items need to be specified about no deviceID(s). It may be made for the value of each line to take other values.

[0059] Already performing single step processing is also considered. That is a terminal attribute rule base is prepared and from the weighted solidity of a terminal like drawing 12 a general terminal attribute is generated rather than using it for conversion of indication structure and generation of a display command. For example when ModemSpeed was smaller than 14.4K and ModemSpeed with slow transmission speed was larger than 56K as except [ with quick transmission speed / its ] was called speed of the degree in the middle. Generating a terminal attributes such as deciding not a concrete numerical value but the range classifying a single parameter or carrying out two or more parameters to an image to a one-dimensional display by there being no pointing device with a slow display speed since transmission speed is slow is also considered.

[0060] Although only the start of the session was explained in the top when a user shows the end of a session clearly and there is no end of an explicit session by that cause it is considered as the end of a session after predetermined time progress from the last access. An end of a session will release the resource used by the data object group and the view object group till then.

[0061] The generation method of the data object is explained briefly here. The contents

creator of Web application draws up the application definition statement document 300 (drawing 13). The application definition statement document 300 is described according to a definition language of a certain kind. This language for example Using XML (W3C Extensible Markup Language (XML) 1.0 and <http://www.w3.org/TR/1998/REC-xml-19980210>) as an original markup language. A definition can be given. In the example of restaurant service an example of description like drawing 14 can be considered.

[0062] Here the restaurant information service which it is going to define now consists of two processes (it is equivalent to the page on a display) of restaurant search and a restaurant information display. Furthermore a restaurant information display process consists of a link for returning to a page of a display item and restaurant search called restaurant information. Similarly it is described that a restaurant information item comprises a display item of the level under it. In drawing 14 the abbreviation is performed selectively. A contents creator may describe creation of such a definition statement document directly using a text editor and an authoring tool like a word processor with a layout function may be used.

[0063] The example which rerun control specifies as the application definition language of drawing 14 is further explained with reference to drawing 15. Further the process of the reservation process of a restaurant is added to the application definition language of drawing 14 so that it may illustrate to drawing 15. It is large and the display item which makes the display item and reservation application which display reservation status input and receives reservation is in this reservation process. It is described to be that these display items belong to the page group "prohibition on rework after reservation acceptance" as an attribute (keyword of pageGrp=). As an attribute of the determination button which fulfills a reservation process when reservation is received it is specified that the display item belonging to this page group is not reworked (keyword of invGrp=).

[0064] Therefore after a user pushes a determination button to the screen for which the display control information generated according to this description was processed by the browser as for these display items and the display item of the level below it rework (reception of a display or re reservation) is forbidden by the mechanism of this invention. even if a user is going to return and do redisplay to the display item (page) of a "reservation status display" with the back button of a browser after reception \*\*\*\* in reservation a display is not performed for the prevention from double reservation.

[0065] Since rerun control of a page corresponds also to the screen generated dynamically it defines a page group using a page definition tree. Above-mentioned drawing 16 is an example of the restaurant of drawing 15 and shows the page definition tree of the portion of a reservation process especially. A reservation process process has three display items of reservation status display reservation acceptance screen and reservation acceptance result screen \*\*. A reservation status display has two kinds of sub display items a reserved date and its empty information. The same can be said for a reservation acceptance screen.

[0066] These display items are dynamically displayed by the screen-display capability of a terminal as a page and it is not decided beforehand which level becomes a page. For example although a reservation status display and a reservation acceptance screen serve as a unit of a page if [ what / whose / has large screen size like PC is / like PDA of a smaller screen ] it may be vacant with the reserved date which is each sub display item and may be

1 page using information.

[0067] Although the page group "prohibition on rework after reservation acceptance" is only specified as the definition of two display items a reservation status display and a reservation acceptance screen in the application definition language of drawing 15 here the child of a page definition tree also belongs to the page group. That is the page group "prohibition on rework after reservation acceptance" has a display item (namely page) in the dotted line of the above-mentioned figure as a member.

[0068] Drawing 13 is referred to again. The definition language-analysis machine 310 analyzes the document written in such a language. As an example of such an analyzer there is IBM XML for Java ([http:// www. alphaworks. ibm.com / formula/xml](http://www.alphaworks.ibm.com/formula/xml)). An analyzer sends an analysis result to a data object generation machine. An analysis result as stated to W3C and "Document Object Model Specification" ([http:// www. w3. org / TR/ WD-DOM-1998-0416](http://www.w3.org/TR/WD-DOM-1998-0416)) It is data expressed by the tree structure (the object of the tag in the inside of nested serves as a child) of an object which made structure of the tag one object.

[0069] A data object generation machine generates a data object based on the tree structure of an analysis result. Generally the class of a data object is determined from the object of an analysis result and an object is generated. Structure also becomes the same form.

[0070] Instead of writing all the information to the application definition statement document the contents of information can also be taken from the existing database. In this case specification of the search method of a database and reference of search results are described by the definition statement document a data object generation machine uses the exchange with a database system called a database connector as an agency mechanism based on these directions and it asks a database. Since the search results of a database are returned through a database connector a data object generation machine generates the data object which reflected the result from a database based on reference directions of search results. The structure at this time is determined by the tag in which the directions to the database of a definition statement document are performed. As an example (drawing 17) of a definition statement document including directions of such a database the case where the restaurant information of drawing 14 is dynamically taken from a database is mentioned.

[0071] The example of the data object group generated by carrying out such processing is shown in drawing 18. The "restaurant detailed page" of which the output is demanded here comprises two objects of "the link to a search page" for returning to "restaurant information" and a retrieval picture. Although restaurant information holds various information about the restaurant which a user wants it is divided into three categories an "attribute" a "guide" and a "menu" here. The object corresponding to information required under each category is generated.

[0072] In this invention the attribute as shown in drawing 19 is defined as the data object. What evaluated the priority of data by considering child objects as the set when child objects existed by a number. Whether even if it divides the data of the object it always displays with a switch (logical value). It is [ whether in the case of an image with a switch whether the split display of the data of this object can be carried out can carry out / outline /-izing and ] a switch in addition arbitrary molds define data peculiar to application if needed. This is the directions information used in generation of a correspondence view

object and generation of said display control information by a view object.

[0073] When this data object is generated two typical examples are shown below about what kind of HTML document what kind of view object is generated and is generated.

[0074] (1) In the example stated to conversion to the display configuration for big screens such as PC and a HTML output destination change take out deviceID and acquire corresponding terminal attribute information from the User-Agent field of HTTP which receives from a client terminal. The following explanation explains as what already acquires this terminal attribute information. In this example terminal attribute information assumes that the information of a "big screen" (the client terminal specifically has a display device of a big screen and resolution according to which it is classified.) is included.

[0075] First the view object corresponding to the object used as the route of the tree structure of a data object is generated (drawing 20). the HTML mechanism of production 230 which performs this -- the class (an example "page class") of a route data object and the peculiar knowledge (for example one restaurant is displayed at 1 page) of the application field -- and An initial parameter when generating the class (class of the page display view object for big screens) of a view object and an object is determined by making terminal attribute information (here "big screen") into a parameter and a route view object is actually generated.

[0076] Thus once the route view object corresponding to a route data object is determined a view object can be made to generate one's one child after another by making this route view object into a starting point. That is the view object has a method requested to generate a child view object in the method which generates the method or the common child view object which generates child objects.

[0077] And if generation of a child view object is requested from the page display view object for big screens It asks the "restaurant detailed display page data object" which is a data object corresponding to itself and its child view object is generated from the child data object. In this generation generally Its own class (here "page display view class for big screens") Generation of an object is performed based on the class of the view object determined from the class of a data object the peculiar knowledge (if it defines) of the application field and terminal attribute information and an initial parameter.

[0078] For example in the case of the "restaurant information" which is a child of a "restaurant detailed display page" data object as shown in drawing 21 From the "general item" and the "big screen" information on terminal attribute information which are the classes of "restaurant information" the "item two-dimensional arrangement display" class is determined as a class of a view object. This item two-dimensional arrangement display view object is a kind of the view object which meant that a terminal screen arranged an output object in two dimensions using a large thing and heightened a display effect. In "the link to a search page" of another child objects of a "restaurant detailed display page" data object the view object that "Link Object" is generated from the class ("hyperlink") in the similar manner.

[0079] If a child view object is generated a parent view object will perform the generation request of the child view object same with he having been requested to all the children. That is as shown in drawing 22 an "attribute list display" view object is generated using that he is a two-dimensional arrangement view object the "attribution information category" of the child data object of "restaurant information" displaying the list of

attributes etc. The "attribute selection display" view object which carries out behavior of choosing one optimal display item from a user's demand or terminal attribute information is generated to the "guide information category" data object which is a data object similarly. To a "menu" data object an item two-dimensional arrangement view object is again generated like restaurant information using the "big screen" of terminal attribute information. In restaurant information the view object which carries out the title display of the restaurant name is added as child objects using the information peculiar to a field that a restaurant name is important as a title.

[0080] Generation of the child view object is similarly requested to these four child view objects "a restaurant name display" an "attribute list display" an "attribute selection display" and "item two-dimensional arrangement of a menu." For example if generation is requested from an "attribute list display" view object the corresponding child data object of an "attribution information category" will be scanned and the view object (attribute name value pair display view object) of the pair of a variable identifier and a value will be generated. In an "attribute selection display" view object. The view object corresponding to one selection (here suppose that the map of an image is chosen from the priority of a data object with PC without a map function) and it for the most suitable thing is generated from the child of a "guide information category" data object. Recursive processing is similarly performed about a menu. The structure of the view object group generated eventually becomes like drawing 23.

[0081] If drawing 23 in which the structure of drawing 18 in which the structure of a data object group is shown here and a view object group is shown is compared. Although the restaurant name view object is added and four child data objects exist under a "guide information category" data object. Since the attribute selection display view object is generated to the "guide information category" data object there is only one child view object of an attribute selection display view object. And the menu name view object is added from importance. As mentioned above as for a data object and a view object structure is not necessarily set to 1 to 1.

[0082] Generation of all the view objects will request generation of HTML from each view object. Namely HTML generation is directed to a route view object. Although the page display view object for big screens generates the control instruction of the HTML page under final output it requests the "item two-dimensional arrangement view object" which is a child and the "hyperlink" object about the contents which are pages. The requested child view object generates the whole HTML requesting its child view object further.

[0083] The outline of the HTML generation by the view object group of drawing 23 is shown in drawing 24. The item two-dimensional arrangement display view object which is a child of the page display view object for big screens generates HTML so that it may be arranged eventually at two dimensions. With reference to terminal attribute information a HTML output is further changeable in the stage which generates this HTML so that it may explain below. In drawing 24 the child view object of an item two-dimensional arrangement display view object is outputting HTML as a graphic display so that two-dimensional arrangement may be carried out. The child view object of an attribute list display view object. About the child view object of two item two-dimensional arrangement display view objects although the Drawings top is not displayed it is outputting HTML so that it may be displayed in the field secured in each parent view

object.

[0084]When there is terminal attribute information which can be interpreted as image display speed being quick enough it is made to output an image as it is on the other hand about the image display view object which is a child view object of an attribute selection display view object (refer to drawing 25). Although the terminal attribute that image display speed is quick enough is determined by transmission speed and drawing throughput it may be made to carry out the above processing because transmission speed is only slow.

[0085]On the other hand when there is terminal attribute information which can be interpreted as image display speed being slow an image display view object outputs the link to an image without outputting an image as it is (refer to drawing 26). This image will be transmitted when a user specifies this image.

[0086]Even if it adopts the method of two-dimensional arrangement in consideration of screen size an aspect ratio etc. it is possible to adjust for every client terminal. Thus the terminal attribute information of the screen of a terminal being "large" and "the display speed of an image being slow" is made to suit and HTML can be generated.

[0087](2) In small screens such as conversion to the display for small screens such as PDA and HTML output PDA since the quantity of the information displayed at a stretch has restriction it is necessary to make the composition and the tree structure of a view object into a thing suitable for it. Terminal attribute information assumes that data which is interpreted as a small screen is included.

[0088]A data object presupposes that it is drawing 18. First the HTML mechanism of production 230 generates the page division display view object for small screens using the class of the route object of a data object and terminal attribute information (in this case a screen is small) like the time of a big screen (refer to drawing 27). Since the route view object was generated a child view object is generated like the time of a big screen.

[0089]However since the amount of information which can be displayed on one screen being restricted this time and information are divided without being shown at once not a view object that performs two-dimensional arrangement but a view object which is displayed in an order from what has the high importance of information is generated.

When this view object generates child objects further a view object is generated so that it may become the sequence structure in consideration of the importance of display information instead of the category structure suitable for two-dimensional arrangement.

[0090]Therefore as shown in drawing 28 corresponding to a restaurant information data object the linearity display view object in consideration of a priority is generated. It is the same as drawing 21 to generate Link Object.

[0091]And when the linearity display view object in consideration of a priority generates a child view object. Something and using sequence structure so that it may be easy to carry out page division are thought as important and what should be promptly shown to a user serves as a completely different tree structure of a view object from the time of a client terminal having a big screen (refer to drawing 29). Although it is generating also except the view object actually displayed at the time in this example it is also possible to make it generate when directed by the user. In drawing 29 it becomes a form where a required child view object is connected to the linearity display view object in consideration of a priority in parallel. The priority is high so that it is close to the linearity display view object which took the priority into consideration by a diagram. Determining

this priority by a certain method reflecting a user's liking is also considered.

[0092]After a view object is generatedHTML is generated sequentially from a route view object like the time of a big screen. The navigation button (NEXT and the PREV button) between pages is also generated automatically so that a user can navigate the divided page easily. The output fitted more to the capability of the terminal can be made to generate using terminal attribute information at this time.

[0093]HTML can be made in the case of drawing 30to generate so that the information on three view objects may be displayed on order with a high priority at 1 page from the information of "screen size 320x240dot" contained in terminal attribute information. Since the screen is still smaller if there is terminal attribute information of "screen size 240x180dot"HTML can also be made to generate so that the information on two view objects may be outputted at 1 page.

[0094]Although Ushiro's screen where the web browser interpreted HTML is expressed with the above figure as an output of a view object groupa view object group outputs the control instruction of HTML.

[0095]Although the above example has explained for HTMLthis invention is not limited to this. For exampleWML. (Wireless Markup LanguageWireless Application Protocol Forum LtdWireless Application Protocol Wireless Markup Language) If outputs, such as Specificationand Version 30-Apr-1998 and XMLare neededWhat is necessary is just to generate structure of a view object group which generates the view object containing a method which outputs themand suits the characteristic of WML or XML.

[0096]Above-mentioned drawing 31 is the figure which explained [ operation of this invention ] the reservation process of the restaurant to the example for time later onin order to explain the mechanism which controls the rerun of a page in relation to processing. Simultaneouslythe contents of the display screen of a terminala user's operation (request content to a server)and the processing state cash in page control DB 213 are also united and displayed. Firstthe reservation status display requirement of a restaurant is emitted from a user. Therebycontrol mechanism 210 of this invention operates and it is distinguished by page control mechanism 212 in it that it is a new page. Nextapplication process mechanism 214 is performed. Herereservation status is investigated by an application program. Simultaneouslythe processing state at this time is stored in state cash in preparation for a rerun. In this processingsince rerun control is unnecessaryscreen creation which is the next processing is performed. As a resultthe display control information for displaying a reservation process situation is sent and displayed on a terminal. A reservation acceptance screen is processed similarly and a state is stored in state cash.

[0097]Nextsuppose that the user tried to do redisplay of the reservation status screen using the history function on a terminal. For examplesuppose that the redisplay button which calls an old reservation status screen to a browser and with which it is similarly usually equipped with the usually equipped back button was pushed. In this casepage control mechanism 212 distinguishes that it is processing to an old screen. In this casesince it is a page in a historyit is confirmed whether the state is stored in state cash. In this casesince it should be stored in state cash as stated abovethat processing state is restored and application is processed (reservation status investigation). Correctlyalthough this result is also stored in state cashin the figureit has excluded for simplification. Finally display control information is generated and it is sent to a terminal.

[0098] Then the above-mentioned drawing 32 explains signs that a rerun is forbidden in the example of the reservation process of a restaurant. A user inputs necessary information into a reservation acceptance screen and suppose that the reservation determination button was pushed. Thereby the control mechanism 210 of this invention operates and it is distinguished by page control mechanism 212 in it that it is a new page. Next application process mechanism 214 is performed. Here reservation acceptance processing is performed by an application program. Simultaneously the processing state at this time is stored in state cash in preparation for a rerun. Since reservation-related rerun control is specified by reservation acceptance in this processing the page to which a rerun is forbidden based on specification "prohibition of the rerun after reservation acceptance" of invGrp in an application definition language is identified. In the case of this example the page of "a reservation status display" and the "reservation acceptance screen" where display item attribute pageGrp of the application description language is specified corresponds. Next the state cash equivalent to these pages is cancelled and screen creation of a reception result is performed.

[0099] Next suppose like a front that the user tried to do redisplay of the reservation status screen using the history function on a terminal. For example suppose that the redisplay button which calls an old reservation status screen to a browser and with which it is similarly usually equipped with the usually equipped back button was pushed. In this case page control mechanism 212 distinguishes that it is processing to an old screen. In this case since it is a page in a history it is confirmed whether the state is stored in state cash. Since unlike the front case state cash is cancelled as stated above it becomes clear that the rerun is forbidden. As a result the display control information which notifies an error is generated it is sent to a terminal and an error picture is displayed on a terminal.

[0100] Above-mentioned drawing 33 explains that the mechanism of rerun control operates satisfactorily in the example of the reservation process of an old restaurant even when screen constitution changes dynamically (terminal attributes). In the old example since the screen was a large terminal the reservation status display was expressed as one screen. Shortly the HTML mechanism of production 230 divides a page supposing the small terminal of a screen for every item of the object "reserved date" of the child of a reservation status display and "empty information." Therefore cash also of the state cash will be carried out for every item of these children. Here like a front a user inputs necessary information into a reservation acceptance screen and suppose that the reservation determination button was pushed.

[0101] Thereby control mechanism 210 of this invention operates and it is distinguished by page control mechanism 212 in it that it is a new page. Next application process mechanism 214 is performed. Here reservation acceptance processing is performed by an application program. Simultaneously the processing state at this time is stored in state cash in preparation for a rerun. Since reservation-related rerun control is specified by reservation acceptance in this processing the page to which a rerun is forbidden based on specification "prohibition of the rerun after reservation acceptance" of invGrp in an application definition language is identified. In the case of this example the page of "a reservation status display" and the "reservation acceptance screen" where display item attribute pageGrp of the application description language is specified corresponds. However since the page of a reservation status display does not exist in cash directly according to a page definition tree the state cash equivalent to these pages is

cancelled using being the "reserved date" which is a child of a reservation status display and a page to which "empty information" corresponds.

[0102]Next suppose that the user was going to be vacant with the reserved date which is one of the reservation status screens and do redisplay of the 1 page of information like the front using the history function on a terminal. For example suppose that the redisplay button which calls an old reservation status screen to a browser and with which it is similarly usually equipped with the usually equipped back button was pushed. In this case the page control mechanism 212 distinguishes that it is processing to an old screen. In this case since it is a page in a history it is confirmed whether the state is stored in state cash. Since state cash is cancelled like the front case as stated above it becomes clear that the rerun is forbidden. As a result the display control information which notifies an error is generated it is sent to a terminal and an error picture is displayed on a terminal.

[0103]In order to specify in the unit of a page group rather than to specify the page actually displayed directly even if the page actually displayed was divided right rerun control demanded with application was able to be carried out.

[0104]

[Effect]The dynamic document or display control information for providing the user interface according to each capability was able to be generated to the client terminal which has various capability.

[0105]The mechanism in which it was made not to depend for the application carried out by a server on the client terminal which uses the processing result of the application concerned was also able to be provided.

[0106]What was necessary was to have been able to change only the creation portion of the display document even if creation of the display document showing processing and the processing result concerned of the application carried out by a server was separated and various client terminals appeared.

[0107]The flexible rerun control and attestation control which were united with processing of application were able to be realized by the easy specification from an application program to the rerun of the page in the history which was with the screen redisplay function of the browser of a terminal. The effective thing was also able to be shown also when describing this specification briefly as a group of a page to description of an application process and performing a dynamic page configuration.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1]It is a figure showing the whole system of this invention.

[Drawing 2]It is a functional block diagram in HTTP server 100.

[Drawing 3]It is a functional block diagram of the control mechanism shown in drawing 2.

[Drawing 4]It is a figure for explaining the process flow of HTTP server 100.

[Drawing 5]It is a figure showing an example of a processing outline.

[Drawing 6]It is a figure for explaining the process flow of Step 1070 shown in drawing 4.

[Drawing 7]It is a figure for explaining the process flow of Step 1072 shown in drawing 6.

[Drawing 8]It is a figure for explaining the process flow of Step 1075 shown in drawing 6.

[Drawing 9]It is a figure showing an example of a processing outline.

[Drawing 10]It is a figure showing an example of a processing outline.

[Drawing 11]It is a figure showing an example of a processing outline.

[Drawing 12]It is a figure showing an example of the contents of terminal attribute information.

[Drawing 13]It is a functional block diagram used when generating a data object.

[Drawing 14]It is a figure showing an example of an application definition statement document.

[Drawing 15]It is a figure showing the example which rerun control specifies as the application definition language of drawing 14.

[Drawing 16]It is an example of the restaurant of drawing 15 and the page definition tree of the portion of a reservation process process is shown especially.

[Drawing 17]It is a figure showing an example of a definition statement document including directions of a database.

[Drawing 18]It is a figure showing an example of a data object group.

[Drawing 19]It is a figure showing an example of the attribute defined as an application object.

[Drawing 20]It is a figure explaining the processing at the time of generating the view object for big screens.

[Drawing 21]It is a figure explaining the processing at the time of generating the view object for big screens.

[Drawing 22]It is a figure explaining the processing at the time of generating the view object for big screens.

[Drawing 23]It is a figure showing the view object group for big screens.

[Drawing 24]It is a figure for explaining the processing at the time of the view object group of drawing 23 outputting HTML.

[Drawing 25]It is a figure for explaining the processing at the time of the view object group of drawing 23 outputting HTML.

[Drawing 26]It is a figure for explaining the processing at the time of the view object group of drawing 23 outputting HTML.

[Drawing 27]It is a figure explaining the processing at the time of generating the view object for small screens.

[Drawing 28]It is a figure explaining the processing at the time of generating the view object for small screens.

[Drawing 29]It is a figure showing the view object group for small screens.

[Drawing 30]It is a figure for explaining the processing at the time of the view object group of drawing 23 outputting HTML.

[Drawing 31]In order to explain the mechanism which controls the rerun of a page in relation to processingit is the figure which explained [ operation of this invention ] the reservation process of the restaurant to the example for time later on.

[Drawing 32]It is a figure which illustrates signs that a rerun is forbiddenin the example of the reservation process of a restaurant.

[Drawing 33]It is an example of the reservation process of an old restaurantand even when screen constitution changes dynamically (terminal attributes)the mechanism of rerun control is a figure explaining operating satisfactorily.

[Explanations of letters or numerals]

100 HTTP server  
110 Network  
120-150 Client terminal  
210 Control mechanism  
220 Data service mechanism  
230 HTML mechanism of production  
240 DB  
250 Terminal DB

---

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2000-82039  
(P2000-82039A)

(43) 公開日 平成12年3月21日 (2000.3.21)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
G 0 6 F 15/00	3 1 0	G 0 6 F 15/00	3 1 0 R
12/00	5 2 0	12/00	5 2 0 E
13/00	3 5 4	13/00	3 5 4 D
17/30		15/40	3 1 0 F
		15/403	3 8 0 Z

審査請求 有 請求項の数20 O L (全 24 頁)

(21) 出願番号 特願平11-38422  
(22) 出願日 平成11年2月17日 (1999.2.17)  
(31) 優先権主張番号 特願平10-184392  
(32) 優先日 平成10年6月30日 (1998.6.30)  
(33) 優先権主張国 日本 (J P)

(71) 出願人 390009531  
インターナショナル・ビジネス・マシー  
ズ・コーポレイション  
INTERNATIONAL BUSIN  
ESS MASCHINES CORPO  
RATION  
アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)  
(72) 発明者 広瀬 紳一  
神奈川県大和市下鶴間1623番地14 日本ア  
イ・ビー・エム株式会社 大和事業所内  
(74) 代理人 100086243  
弁理士 坂口 博 (外1名)

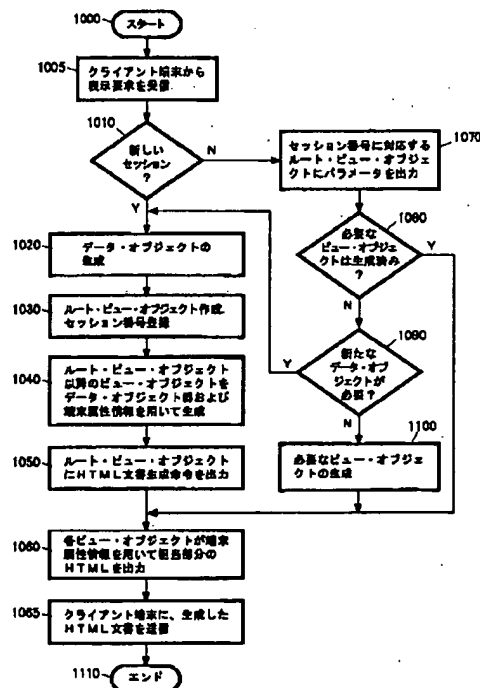
最終頁に続く

(54) 【発明の名称】 表示制御情報生成方法及びコンピュータ

(57) 【要約】

【課題】 様々な能力を有するクライアント端末に対し、それぞれの能力に応じたユーザ・インターフェースを提供するための動的な文書又は表示制御情報を生成すること。

【解決手段】 ユーザが使用する端末に適合した形態で表示を行うように表示制御情報 (例えば HTML) を生成する方法であって、端末から表示要求を受信するステップと、表示要求に応答して、表示内容に関する、端末に独立のデータ・オブジェクト群を生成するステップと、データ・オブジェクト群及び端末属性情報を用いて、表示制御情報を生成するためのビュー・オブジェクト群を生成するステップと、端末属性情報を参照して、ビュー・オブジェクト群により表示制御情報を生成するステップとを含む。



## 【特許請求の範囲】

【請求項1】ユーザが使用する端末に適合した形態で表示を行うように表示制御情報を生成する方法であって、前記端末から表示要求を受信するステップと、前記表示要求に応答して、表示内容に関する、前記端末に独立のデータ・オブジェクト群を生成するステップと、

前記データ・オブジェクト群及び端末属性情報を用いて、前記表示制御情報を生成するためのビュー・オブジェクト群を生成するステップと、前記端末属性情報を参照して、前記ビュー・オブジェクト群により前記表示制御情報を生成するステップと、を含む表示制御情報生成方法。

【請求項2】前記端末から端末識別情報を受信するステップと、前記端末識別情報から前記端末属性情報を取得するステップと、をさらに含む請求項1記載の表示制御情報生成方法。

【請求項3】前記ビュー・オブジェクト群を生成するステップが、前記ビュー・オブジェクト群のルート・オブジェクトを生成するステップと、前記ルート・オブジェクトにより子オブジェクトを生成するステップと、を含む請求項1記載の表示制御情報生成方法。

【請求項4】前記データ・オブジェクトが、対応ビュー・オブジェクトの生成及びビュー・オブジェクトによる前記表示制御情報の生成において用いられる指示情報を保持している、請求項1記載の表示制御情報生成方法。

【請求項5】前記端末属性情報が、前記端末の画面の大きさ情報を含む、請求項1記載の表示制御情報生成方法。

【請求項6】前記表示制御情報が、ハイパー・テキスト・マークアップ・ランゲージ（HTML）である、請求項1記載の表示制御情報生成方法。

【請求項7】前記表示制御情報を生成するステップが、前記ビュー・オブジェクト群のルート・オブジェクトから順に担当部分の表示制御情報を、前記端末属性情報を参照して、生成するステップを含む請求項1記載の表示制御情報生成方法。

【請求項8】前記端末からの表示要求に応答して、当該表示要求により前記データ・オブジェクト群を生成する必要があるか判断するステップと、前記データ・オブジェクト群を生成する必要がある場合には、前記ビュー・オブジェクト群中の関連するビュー・オブジェクトが前記表示要求に対応する表示制御情報を生成するステップと、をさらに含む請求項1記載の表示制御情報生成方法。

【請求項9】履歴上のページの再実行・再認証の制御を行う方法であって、

ページ定義をグループ化するステップと、グループ化したページ定義を前記表示制御情報の生成に用いられる指示情報に含めるステップと、履歴上のページの再実行が行われたかどうかを判別するステップと、再実行されていてかつ処理状態のキャッシュが存在する場合は処理状態を復元するステップと、アプリケーションの処理の実行とそのときの処理状態をキャッシュに格納するステップと、アプリケーションの処理の結果に応じて前記ページ定義のグループを指定して該当するページに対応する処理状態をクリアするステップとを含む請求項1記載の表示制御情報生成方法。

【請求項10】履歴上のページが認証が必要かどうかと認証状態をチェックするステップと、認証が必要なときに認証を確認するステップと、アプリケーションの処理の結果によって履歴上のページの認証状態を変更するステップとをさらに含む請求項9記載の表示制御情報生成方法。

【請求項11】ユーザが使用する端末に適合した形態で表示を行うように表示制御情報を生成するコンピュータであって、前記端末からの表示要求に応答して、表示内容に関する、前記端末に独立のデータ・オブジェクト群を生成するモジュールと、前記データ・オブジェクト群及び端末属性情報を用いて、前記表示制御情報を生成するためのルート・ビュー・オブジェクトを生成するモジュールと、前記ルート・ビュー・オブジェクトに、その子ビュー・オブジェクトを生成するように命ずるモジュールと、前記ルート・ビュー・オブジェクトに前記表示制御情報の生成を命ずるモジュールと、を有するコンピュータ。

【請求項12】前記端末から受信した端末識別情報から前記端末属性情報を導出するモジュール、をさらに有する請求項11記載のコンピュータ。

【請求項13】前記子ビュー・オブジェクトが、その子オブジェクトを生成することができる、請求項11記載のコンピュータ。

【請求項14】前記ルート・ビュー・オブジェクトが、前記表示制御情報の生成を前記子ビュー・オブジェクトに命ずることができる、請求項11記載のコンピュータ。

【請求項15】前記ルート・ビュー・オブジェクト及び前記子ビュー・オブジェクトを含むビュー・オブジェクト群は、前記端末属性情報を参照して前記表示制御情報を生成する、請求項11記載のコンピュータ。

【請求項16】前記端末からの表示要求に応答して、当該表示要求により前記データ・オブジェクト群を生成す

る必要があるか判断し、前記データ・オブジェクト群を生成する必要がある場合には、前記ルート・ビュー・オブジェクトに前記表示要求を出力するモジュール、を有する請求項1記載のコンピュータ。

【請求項17】履歴上のページの再実行・再認証の制御を行うコンピュータであって、ページ定義をグループ化するモジュールと、グループ化したページ定義を前記表示制御情報の生成に用いられる指示情報に含るモジュールと、履歴上のページの再実行が行われたかどうかを判別するモジュールと、再実行されていてかつ処理状態のキャッシュが存在する場合は処理状態を復元するモジュールと、アプリケーションの処理の実行とそのときの処理状態をキャッシュに格納するモジュールと、アプリケーションの処理の結果に応じて前記ページ定義のグループを指定して該当するページに対応する処理状態をクリアするモジュールとを有する請求項1記載のコンピュータ。

【請求項18】履歴上のページが認証が必要かどうかと認証状態をチェックするモジュールと、認証が必要なときに認証を確認するモジュールと、アプリケーションの処理の結果によって履歴上のページの認証状態を変更するモジュールとをさらに有する請求項17記載のコンピュータ。

【請求項19】ユーザが使用する端末に適合した形態で表示を行うように表示制御情報をコンピュータに生成させるプログラムを記憶した記憶媒体であって、前記プログラムは、前記コンピュータに、前記端末からの表示要求に回答して、表示内容に関する、前記端末に独立のデータ・オブジェクト群を生成するステップと、前記データ・オブジェクト群及び端末属性情報を用いて、前記表示制御情報を生成するためのルート・ビュー・オブジェクトを生成するステップと、前記ルート・ビュー・オブジェクトに、その子ビュー・オブジェクトを生成するように命ずるステップと、前記ルート・ビュー・オブジェクトに前記表示制御情報の生成を命ずるステップと、を実行させる、記憶媒体。

【請求項20】履歴上のページの再実行・再認証の制御をコンピュータに行わせるプログラムを記憶した記録媒体であって、前記プログラムは、前記コンピュータに、ページ定義をグループ化するステップと、グループ化したページ定義を前記表示制御情報の生成に用いられる指示情報に含るステップと、履歴上のページの再実行が行われたかどうかを判別するステップと、再実行されていてかつ処理状態のキャッシュが存在する

場合は処理状態を復元するステップと、アプリケーションの処理の実行とそのときの処理状態をキャッシュに格納するステップと、アプリケーションの処理の結果に応じて前記ページ定義のグループを指定して該当するページに対応する処理状態をクリアするステップとを実行させる請求項19記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ユーザが使用する端末に適合した形態で文書等の表示を行うための制御命令を生成する方法及び機構に関し、より詳しくは、ユーザが使用する様々な端末、例えばPDA(Personal Digital Assistance)や通常のパーソナル・コンピュータ(PC)、テレビ、又は情報表示機能を有する携帯電話等の表示画面の大きさや通信速度等に適合した形態で、HTML(Hyper Text Markup Language)等を出力する方法及び機構に関する。

【0002】さらに、本発明は前記の動的にページを生成する機構に関連して、端末の履歴機能を考慮して認証や再実行を細かく制御する機構に関するものである。より詳しくは、データベースの更新などのトランザクションや厳密なユーザ認証が必要なページの表示を行う際に、ページが動的に生成されるような場合でも、再実行の抑制や再認証などの処理がアプリケーションから簡単に指定でき、動作するような機構に関する。

【0003】

【従来の技術】Webベースの技術の浸透に伴い、ユーザ・インターフェースをWebブラウザを用い、アプリケーション本体はWebサーバ(アプリケーション・サーバ)で稼働させるような、いわゆるシン(thin)クライアント型のシステムが利用されるようになってきている。この時、サーバからクライアントに送られる情報はフォームを含むHTML文書であり、クライアントの端末では、Webブラウザさえ動けば、実行に先立って特にアプリケーションの配布・インストール等の作業を行うことなく、アプリケーションを実行させることができるようになる。また、一般的には、クライアント側でアプリケーションを実行するのに比して、クライアント側はより処理能力の低いハードウェアで運用することができる。

【0004】従来、このようなクライアント端末には、ローエンドのPCが用いられてきた。しかし、最近では、更に処理能力の低く且つ低コストな、PDA、ハンドヘルドPC、セット・トップ・ボックス、そしてカーナビゲーション・システム等でもWebブラウザが動くようになってきており、またWebブラウザ機能を有する携帯電話のような新しい端末機器も開発されてきている。そのため、例えば、広い表示領域を有するPCではうまく表示されていたWebページが、より狭い表示領

域を有するPDAでは頻繁なスクローリング操作を必要とするようになるなど、使い勝手が非常に悪くなってしまうという問題が生じた。

【0005】この問題に対処する方法としては、特開平10-133973号に記載されているように、対象となる全ての種類のクライアント端末に対応した複数のHTML文書とそれらに付随するその他のリソースを予め用意しておき、クライアントからのリクエストごとに、端末の種類を判別して適当なものを返すという方法がある。しかし、この場合、大量のデータを作成する必要があり、システム開発のワークロードが膨大になるという欠点がある。また、サポートすべきクライアント端末の種類が増えた時のワークロードも大きい。

【0006】また他の方法としては、PC向けの標準的なHTML文書のみを用意し、クライアント端末の能力に合わせて、その内容を変更するという方式も提案されている。これは、例えば、表示領域が狭い端末に対しては、縮小した画像を作成し、クライアントにはその小さい画像を参照するように変更されたHTML文書を送るものである。先に述べた特開平10-133973号には、予め用意されたHTML文書を何等かの手段で、複数の端末に適合できるように変形するという事項も記載されている。しかし、変形についての具体的方法は記載されていない。

【0007】また特開平10-21165号には、サーバから取得したハイパーテキストのデータサイズやデータ配置を、それを受信するクライアントの物理環境やユーザ生活シーン等に合わせて編集して送信する通信システムが開示されている。この物理環境には、通信回線種別や、ワークステーション又は携帯電話などの端末の性能、画像の色数やサイズ等を含む。またユーザ生活シーンは、出勤中、主張中、在籍中、休日等のデータを含むものである。本公報では、送信される情報の取捨選択に物理環境やユーザ生活シーン等を用いることが開示されており、画面構成については特に開示されていない。

【0008】さらに、T.W.Bickmore et al, "Digester: Device-Independent Access To TheWorld Wide Web,"

(<http://www.fxpai.xerox.com/papers/bic97>) には、プロキシ(proxy)が、HTML文書のリクエストを受け取ると、特定されたHTTP(Hyper Text Transport Protocol)サーバから文書を取り出し、そのHTML文書をパースし、AST(Abstract Syntax Tree)を作成し、このASTに所定の変形操作を施すことにより新たなHTML文書を作成することを示している。このDigesterでは、HTTPサーバは最初にHTML文書を作成しなければならず、当然ASTを作る際にもそのHTML文書が必要とされる。また、変形操作が実施されたASTはそのままHTML文書となるため、AST自体はHTML文書を作成する際の制御を実施するわけではない。

【0009】あるいは、ページの再実行に関しては、URLやHTTPヘッダ情報(いわゆるCookies)にセッションの識別情報や状態を付加することが可能である(例えば、Servletの機能(SUN, Java Servlet API Specification Ver. 2.1)、IBM WebSphereのSession Trackingなどを参照)。しかし、これらは状態の識別と復元をするだけあり、本発明の目的とするページの再実行の制御は含まれていない。

【0010】データベースのトランザクション処理におけるページの再実行制御に関しては、URLに番号をつけて、一度表示されたページの再実行を禁止するというテクニックは広く知られている(例えば、RFC2068 HTTP1.1 Specificationの13.13 History Listsに"once-only" URLとして紹介されている)。

【0011】しかし、認証状態や再実行制御を、アプリケーションの処理や端末の多様性によって、動的に変えようとするとプログラミングが複雑になる(基本的な機能のみでは柔軟な処理は不可能で、本来は再実行できるようところもすべて再実行できなくなったりする)。また、ページが組織化されていないので、このような処理プログラムのページの指定が複雑になり、ページの追加や削除などがおこると全体を書き換える必要が生じ、ソフトウェアの生産性・保守性や信頼性が低下する。端末によってページを自動的に分割するようなシステムではこのような方式で制御することは困難である。

【0012】

【発明が解決しようとする課題】本発明は、様々な能力を有するクライアント端末に対して、それぞれの能力に応じたユーザ・インターフェースを提供するための動的な文書又は表示制御情報を生成することである。

【0013】また、サーバで実施されるアプリケーションを、当該アプリケーションの処理結果を使用するクライアント端末に依存しないようにする機構を提供することも目的である。

【0014】さらに、サーバで実施されるアプリケーションの処理と当該処理結果を表す表示文書の作成とを分離し、様々なクライアント端末が出現しても、表示文書の作成部分のみを変更すればよいようにすることである。

【0015】本発明は、さらに、端末上の履歴機能を用いた過去のページ上への操作と要求に対して、その再実行や再認証を制御するしくみと、アプリケーションプログラムからその制御内容を簡単に指定するしくみを提供することを目的とする。

【0016】また、この制御するしくみは、端末にあわせて、あるいは、アプリケーションによってページが動的に生成されるときも、動作可能とすることを目的とする。さらに、アプリケーションプログラムから、ページ

の動的な生成やアプリケーションによる影響を排除して、簡単に前記の制御を指定できるようにすることを目的とする。

【0017】

【課題を解決するための手段】本発明は、ユーザが使用する端末に適合した形態で表示を行うように表示制御情報（例えばHTML）を生成する方法であって、端末から表示要求を受信するステップと、表示要求にตอบสนองして、表示内容に関する、端末に独立のデータ・オブジェクト群を生成するステップと、データ・オブジェクト群及び端末属性情報を用いて、表示制御情報を生成するためのビュー・オブジェクト群を生成するステップと、端末属性情報を参照して、ビュー・オブジェクト群により表示制御情報を生成するステップとを含む。このようにデータ・オブジェクト群は、端末とは独立に生成されるものであるから、新たな端末が表示要求を出力するようになったとしても、データ・オブジェクト群を生成する部分及び要求内容が同じであればデータ・オブジェクト群自身も変更は必要ない。なお端末は、通常のPC、PDA、ブラウザを使用可能なTVや携帯電話等の機器である。

【0018】また、端末から端末識別情報を受信するステップと、端末識別情報から端末属性情報を取得するステップとをさらに含むようにすることもできる。端末属性情報の取得には様々な方法がある。このように端末識別情報と端末属性情報を対にして予め記憶しておく方法と、端末属性情報を最初にアクセスした際に入力する方法等がある。

【0019】先に述べたビュー・オブジェクト群を生成するステップは、ビュー・オブジェクト群のルート・オブジェクトを生成するステップと、ルート・オブジェクトにより子オブジェクトを生成するステップとを含むようにすることも可能である。ルート・オブジェクトは端末属性情報を参照して子オブジェクトを生成するメソッドを有することもある。この子オブジェクトは、必要に応じて孫オブジェクトを生成する。

【0020】データ・オブジェクトが、対応ビュー・オブジェクトの生成及びビュー・オブジェクトによる表示制御情報の生成において用いられる指示情報を保持するよう構成することができる。指示情報とは、オブジェクトの優先順位や、分割可否情報、アウトライン可否情報等を含む。また、端末属性情報は、端末の画面の大きさ情報や、通信速度、画面の縦横比、又は色数の情報である。さらに、表示制御情報は、実施例ではハイパー・テキスト・マークアップ・ランゲージ（HTML）であるが、その他のマークアップ・ランゲージでもよい。

【0021】先に述べた表示制御情報を生成するステップは、ビュー・オブジェクト群のルート・オブジェクトから順に担当部分の表示制御情報を、端末属性情報を参照して、生成するステップを含むようにすることも可能

である。ルート・オブジェクトは、ページ全体に関する表示制御情報を生成し、子オブジェクトはそのページの内容についての表示制御情報を生成する。なお、ビュー・オブジェクトは端末属性情報を参照するので、このビュー・オブジェクトが同じでも異なる表示制御情報が生成されることもありえる。

【0022】本発明では、同じ端末からの表示要求であって以前の表示要求に関連するものである場合には、端末からの表示要求にตอบสนองして、当該表示要求によりデータ・オブジェクト群を生成する必要があるか判断するステップと、データ・オブジェクト群を生成する必要がある場合には、ビュー・オブジェクト群中の関連するビュー・オブジェクトが表示要求に対応する表示制御情報を生成するステップとをさらに実行する場合がある。データ・オブジェクトに変更が必要なければ、ビュー・オブジェクトが表示制御情報を新たに作成することにより対処することができる。但し、データ・オブジェクト群を新たに作成せずとも、必要なビュー・オブジェクトを生成しなければならない場合もある。

【0023】以上のように、データ・オブジェクトもビュー・オブジェクトも必要に応じて生成されるので、これらを生成する前にサーバに存在するのは、端末からの表示要求にตอบสนองして、表示内容に関する、端末に独立のデータ・オブジェクト群を生成するモジュールと、データ・オブジェクト群及び端末属性情報を用いて、表示制御情報を生成するためのルート・ビュー・オブジェクトを生成するモジュールと、ルート・ビュー・オブジェクトに、その子ビュー・オブジェクトを生成するように命ずるモジュールと、ルート・ビュー・オブジェクトに表示制御情報の生成を命ずるモジュールである。

【0024】端末から受信した端末識別情報から端末属性情報を導出するモジュールをさらに含む場合がある。

【0025】上で述べた子ビュー・オブジェクトが、その子オブジェクトを生成するメソッドを有しており、ルート・ビュー・オブジェクトが、表示制御情報の生成を子ビュー・オブジェクトに命ずるように構成することも可能である。また、ルート・ビュー・オブジェクト及び子ビュー・オブジェクトを含むビュー・オブジェクト群は、端末属性情報を参照して表示制御情報を生成するメソッドを有するように構成することもできる。

【0026】なお、以上説明した本発明をプログラムにて実装した場合には、当該プログラムはCD-ROM又はFD等の記憶媒体に格納される場合がある。

【0027】履歴上のページの再実行や再認証などの制御を行うために、アプリケーションの開発時にページ定義をグループ化する。このグループはさらに別のサブグループを含んでもよく、ページ定義全体はツリー構造を形成する。このページ定義は、前記表示制御情報の生成に用いられる指示情報に含めることができる。この指示情報内のページ定義に対してグループ化を行う。これに

より、ページの任意のグループ名を指定するだけで、後で述べる再実行が禁止されるページの集合や認証状態が変更されるページの集合を簡単に、かつ、柔軟に指定することができる。

【0028】実行時には、前記の表示制御情報の生成ステップに加えて、履歴上のページの再実行が行われたかどうかを判別するステップ、再実行されていてかつ処理状態のキャッシュが存在する場合は処理状態を復元するステップ、アプリケーションの処理の実行とそのときの処理状態をキャッシュに格納するステップ、アプリケーションの処理の結果に応じて前記ページ定義のグループを指定して該当するページに対応する処理状態をクリアするステップ、を統合することができる。

【0029】さらに、前記の表示情報生成のステップと履歴上のページの再実行制御のステップに加えて、履歴上のページが認証が必要かどうかと認証状態をチェックするステップ、認証が必要ときに認証を確認するステップ、アプリケーションの処理の結果によって履歴上のページの認証状態を変更するステップ、を統合することができる。ただし、認証状態の変更は、再実行の制御と同じようにページ定義のグループを指定することで簡単にアプリケーション開発者が指示することができる。

#### 【0030】

【発明の実施の形態】まず、本発明に関するシステム全体を図1を用いて説明する。図1には、本発明に関連するWebサーバ100と、通常のPC120、テレビ130、携帯型コンピュータ140、及びWebブラウザを内蔵する携帯電話150がネットワーク110にて接続されている状態を示している。PC120、テレビ130及び携帯型コンピュータ140はWebブラウザを有しており、HTTPサーバ100から情報を取得する。クライアント端末は、Webブラウザを有する他の機器でもよい。なお、携帯電話150は無線を介してネットワーク110に接続しているため図1では点線でネットワーク110に接続している。このように、ネットワーク110に接続するコンピュータは、それぞれ表示画面の大きさや、通信速度、処理能力、操作性が異なっている。よって、各々の表示装置に表示される文書は、それぞれの端末の能力に適合したものであることが好ましい。表示する文書を端末に適合させる方法には、端末側で対処する方法もあるが、本発明ではHTTPサーバ100を以下に説明するような構成とする。

【0031】ではこのHTTPサーバ100の機能ブロック図を図2に示す。また、HTTPサーバ100内の機能ブロック内の制御機構210のサブ機能の詳細ブロック図を図3に示す。

【0032】図2に示すように、HTTPサーバ100は、制御機構210と、DB240と、データ・サービス機構220と、HTML生成機構230と、端末DB250を含む。DB240を含まない場合もある。

【0033】制御機構210は、クライアント端末120乃至150からの各種パラメータを受信して、それに対応する適当なHTML文書を返信する。このために、データ・サービス機構220及びHTML生成機構230の動作を制御する。Webアプリケーションでは、同じクライアント端末から一連のリクエストをもって処理の一単位とすることが多い。そのため、複数のリクエストの中から、同じクライアントからのものを識別したり、HTML文書の生成が終了しても、すぐにはリソースを開放せずに状態を保持して次のアクセスに備えることによりアプリケーションとしての実行を継続する必要がある。

【0034】ここでは、クライアント端末による一連のアクセスをセッションと呼ぶ。このため制御機構210は、セッションを管理し、あるリクエストが既存のセッションに属するものなのか、新たなセッションを開始すべきなのかを判断する。また、クライアント端末120乃至150から得られる識別情報から端末属性情報を得るために端末DB250にアクセスし、その端末属性情報をHTML生成機構230に出力するようにしてもよい。データ・サービス機構220及びHTML生成機構230はそれぞれが端末DB250にアクセスするようにしてもよい。

【0035】図3に示すように、制御機構210はページの再実行や認証などの制御をするために、セッション識別機構211、ページ管理機構212、ページ管理DB213、アプリケーション実行装置214、ページ再実行制御装置215の詳細化された機能から構成される。

【0036】セッション識別機構211は端末から送付されるパラメータを解析してどのセッションが端末からリクエストしているかを識別し、セッションに固有のデータを同定する。同様にパラメータを解析して、ブラウザの履歴上のページが再実行されているかどうかやそのページが認証が必要かどうかを判断してページ管理機構212が必要なページを識別し、ページ管理DB213から必要な情報を取り出し認証や再実行に必要な過去の処理状態の復元を行う。ページ管理機構212の処理が終わるか、このような処理が必要なければ、アプリケーション実行装置214に処理が移る。

【0037】アプリケーション実行装置214はWebアプリケーションに固有の処理を実行する。例えば、バンキングアプリケーションの場合は実際の銀行取引が行われる。この処理結果をもとに、ページ再実行制御装置215がページ管理DBを更新する。すなわち、処理が過去のページの再実行を禁止するような場合はページ管理DB213から処理の再実行に必要な状態キャッシュを取り除く。

【0038】データ・サービス機構220は、クライアント端末120乃至150からの検索要求等の処理要求を実施するため、必要に応じてDB240にアクセス

し、データ・オブジェクト群を生成する部分と、DB 240へアクセスしその内容を表現し且つセッションの処理を実行するデータ・オブジェクト群自体とを含む。このデータ・オブジェクト群は、セッションごとに生成され、本実施例では、Webページの文書構造に従って、処理プロセス全体を表すオブジェクトをルートとした木構造として構成される。葉にあたるオブジェクトは文字列や画像等のデータに対応するものとなる。但し、木構造に限定されるものではない。

【0039】HTML生成機構230は、データ・オブジェクト群及び端末DB 250からの情報を用いて、ビュー・オブジェクト群を生成する部分と、ビュー・オブジェクト群自体とを含む。ビュー・オブジェクト群は、ルート・ビュー・オブジェクトが制御機構210からのHTML生成命令を受信すると、ルート・ビュー・オブジェクトから順にHTMLを出力する。このHTML出力の際にも、端末DB 250から得られた情報は使用される。データ・オブジェクト群から生成されるビュー・オブジェクト群の構造は、端末DB 250から得られる情報に基づき変化し、またビュー・オブジェクト群により出力されるHTML文書も同様に变化する。すなわち、同じデータ・オブジェクト群が生成されても、ビュー・オブジェクト群の構造は変化し得るし、ビュー・オブジェクト群が同一でも出力されるHTML文書は異なる場合がある。HTML文書全体が生成されると制御機構210から要求元のクライアント端末120乃至150に送信される。なお、ビュー・オブジェクト群を生成する部分及びビュー・オブジェクト群自身は後に述べる他のパラメータを参照するようにすることも可能である。

【0040】現在のHTTPサーバ100において、動的なHTML文書の生成は、CGI (Common Gateway Interface) 又はサプレットという仕組み (Sun, Java, JDK1.2, "http://jserv.javasoft.com/products/java-server/servlets/") を用いて行っている。以下ではサプレットを用いて実現する例を述べる。これは、Webアプリケーションに必要とされる継続的なデータ保持が容易であり、以下で述べる本発明に必要な要件を満たしているからである。但し、CGIで実現することも可能である。その場合にはCGIで起動されるプロセスとプロセス間通信を行うようなバックグラウンドのプロセスを用いる。また、本発明に必要な構成要素を、Web代理サーバという形で実現することも可能である。

【0041】では、HTTPサーバ100の処理全体を図4を用いて説明する。まず、クライアント端末120乃至150から表示要求を受信する (ステップ1005)。この表示要求には、例えばURL (Uniform Resource Locator) 及びHTTPのヘッダに含まれるUser-Agentフィールドを用いる。例えば、URLは、"http://aServer/servlet/RestaurantDemo"であり、HTTPの

ヘッダのUser-Agentフィールドは"deviceId"となる。なお、User-Agentフィールドは後に用いる端末属性情報を端末DB 250から得るのに用いる。そして、URLから新しいセッションかどうか判断する (ステップ1010)。新たなセッションである場合には、URL内にsession=<セッション番号>が入れていないので、ステップ1020に移行する。一方、新たなセッションでない場合には、URL内にはsession=<セッション番号>が入れているので、そのセッション番号を用いる。

【0042】新しいセッションの場合、まずデータ・アクセス機構220が要求された表示内容に関するデータ・オブジェクト群を生成する (ステップ1020)。このデータ・オブジェクトの生成処理は後に述べる。

【0043】次に、ルート・ビュー・オブジェクトをHTML生成機構230により生成する (ステップ1030)。このビュー・オブジェクトは、子ビュー・オブジェクトを作成するメソッド (又は子ビュー・オブジェクトを生成するように、子ビュー・オブジェクト生成機構に依頼するメソッド) とHTMLを生成するメソッドを有している。そして、ルート・ビュー・オブジェクトとセッション番号の対をテーブルに登録する。このテーブルは、ステップ1010において用いられる。そして、ルート・ビュー・オブジェクト以降のビュー・オブジェクト群を、生成されたデータ・オブジェクト群及び端末属性情報を用いて、順次生成していく。この処理については後に例を用いて説明する。必要なビュー・オブジェクト群が生成されると、ルート・ビュー・オブジェクトに制御機構210からHTML生成命令が出力される

(ステップ1050)。これを受け取ったルート・ビュー・オブジェクトは、担当分のHTMLを生成し、以下自分の子ビュー・オブジェクトにHTML生成を依頼していく。そうすると、子ビュー・オブジェクト以下のビュー・オブジェクトは自分の担当分のHTMLを生成する (ステップ1060)。そして、制御機構210は、生成されたHTML文書をクライアント端末に送信する (ステップ1065)。

【0044】ここまでの処理を模式的に示したものが図5である。まず、HTTPのUser-AgentフィールドとURLを制御機構210が受信する。そして、データ・サービス機構220は、制御機構210から受け取ったURLの情報からデータ・オブジェクト群を生成する。一方、User-Agentフィールドは、端末属性情報を取得するのに用いられる。そして、制御機構210は、ルート・ビュー・オブジェクトを生成するようにHTML生成機構230に命じ、HTML生成機構230はデータ・オブジェクト群のルート・オブジェクト及び端末属性情報を参照しつつ、ルート・ビュー・オブジェクトを生成する。ルート・ビュー・オブジェクト以外もデータ・オブジェクト群及び端末属性情報を参照しながら、親ビュー・オブジェクトが子ビュー・オブジェクトを生成してい

く。生成されたルート・ビュー・オブジェクトに制御機構210からHTML生成命令が出されると、ルート・ビュー・オブジェクト以降の各ビュー・オブジェクトにより、図5に示されたようなHTML文書が生成され、クライアント端末に出力される。

【0045】では、新しいセッションでない場合、セッション番号がURLに存在することになるので、セッション番号に対応するルート・ビュー・オブジェクトに受け取ったパラメータを出力する(ステップ1070)。そして、URLのセッション番号以外のパラメータから、必要なビュー・オブジェクトが生成済みであるかどうか判断する(ステップ1080)。この判断は、ルート・ビュー・オブジェクト自身が行っても、ルート・ビュー・オブジェクトより下のビュー・オブジェクトがパラメータを受け取った時点で行ってもよい。もし、必要なビュー・オブジェクトが生成されていれば、ステップ1060(又はステップ1050)に移行する。すなわち、各ビュー・オブジェクトが端末属性情報を用いて担当部分のHTMLを出力し、クライアント端末に生成したHTML文書を送信する。

【0046】ステップ1070を、図6を参照してさらに説明する。ステップ1070は、ページの認証や過去のページの再実行制御などのページ制御のために、さらにサブステップにわけられる。まず、端末から送信されたパラメータを解析することにより、ページの制御が必要かどうかを判別(ステップ1071)し、必要な場合はページ管理制御を行う(1072)。例えば、ブラウザの履歴上のページをユーザが再実行した場合、リクエストのURLやHTTPヘッダにページ識別子を埋め込んでおくことにより、過去のページが再実行されたことがわかるので、その識別子をキーにページ管理DBからそのページが表示されたときの処理状態を復元する。また、同様に認証が必要なページには認証のトークンをURLやHTTPヘッダに埋め込んでおくことにより、認証を行うことができる。

【0047】ステップ1072を、図7を参照してさらに説明する。ステップ1072のページ管理制御として、具体的には、図7に示すように、ページの認証や過去のページの再実行準備が行われる。上記の図はこれらのサブステップを表したものである。まず、端末から送信されたパラメータを解析することにより、ブラウザの履歴上のページが使用されたかを判別する(ステップ107201)。ここで、過去の履歴でない新しいページが使用され得る場合は、このサブフロチャートの処理対象外であるので、そのまま次の処理へ抜ける。過去の履歴上のページである場合は、ページ管理DB 213を参照して、過去の処理状態のキャッシュが存在するか調べる(ステップ107202)。キャッシュが存在しない場合は再実行が禁止されているということなので、該当するエラー処理(ステップ107206)にとぶ。キャッシュが

存在する場合は、それを利用して処理状態を復元する(ステップ107203)。次に、認証が必要なページかどうかを、ページのツリー構造を用いて、判別する(ステップ107204)。この判別はそのページが認証を必要とするページグループに属しているかどうかをページのツリー構造を利用して調べることにより行われる。認証が不用の場合は、そのままサブフロチャートの次の処理へ抜ける。認証が必要である場合は、端末から送信されたパラメータに含まれる認証トークンやページ管理DB 213でそのページの認証が無効化されていないかどうかをチェックして認証が有効かどうかを判断する(ステップ107205)。認証が有効なら、そのまま次の処理へ抜ける。無効の場合はステップ107206のエラー処理へとぶ。

【0048】再び、図6を参照する。次に、アプリケーション本来の処理を実行する(1073)。例えば、バンキングのアプリケーションなら指定された銀行取引を行う。この時、ブラウザの履歴ページの再実行に備えてページ管理DBにアプリケーション実行後の状態をキャッシュしておく。さらに、このアプリケーションの処理結果をもとにページ制御が変更されたかどうかを判別(1074)し、必要ならページ管理DBを更新する(1075)。例えば、データベースのコミットが行われた場合はデータベースの更新処理のページの再実行を禁止する必要がある。すなわち、ページ管理DBの該当する処理状態キャッシュをクリアする。別の例として、処理の結果認証が行われた(あるいはそれまでの認証が破棄された)場合は、ページ管理DBの該当部分の更新が行われる。

【0049】ステップ1075を、図8を参照してさらに説明する。ステップ1075のページ管理DB更新として、具体的にはページの認証の無効化や過去のページの再実行禁止が行われる。上記の図はこれらのサブステップを表したものである。まず、ステップ1075の前のステップであるアプリケーション実行(ステップ1073)の結果を調べてその実行結果によりあるページグループの再実行を禁止するかどうかを判別する(ステップ107501)。もし、再実行を禁止する必要があるれば、アプリケーションプログラムで指定されたページグループ名と、ページ定義のツリー構造をもとにページグループを算出する(ステップ107502)。次に、該当するページグループに属するページに対応する処理状態のキャッシュをページ管理DB 213から取り除く(ステップ107503)。再実行を禁止する必要がなければ、ステップ107502、107503はスキップされる。同様にアプリケーションの処理結果により、認証が無効化されるかどうかを判別する(ステップ107504)。認証を無効化する必要があれば、同様にページ定義のツリー構造をもとに認証が無効化されるページグループを算出する(ステップ107505)。ステップ107506では、該当するページグループに属するページの認証を無効化する。認証を無効化する必

要がなければ、ステップ 107505, 107506 はスキップされる。

【0050】再び図4を参照する。一方、必要なビュー・オブジェクトが生成されていないと判断された場合には、新たなデータ・オブジェクトが必要であるか判断する(ステップ1090)。新たなデータを必要とするような場合には、それ以前に生成されたデータ・オブジェクトではそのデータを得ることができない場合もあるので、必要な場合にはデータ・サービス機構220に、データ・オブジェクトを生成するように依頼する(ステップ1020への移行)。なお、同じセッションであるが新たなデータ・オブジェクトが必要になった場合には、元のデータ・オブジェクトのサブ・ツリーを生成することになるので、ステップ1030におけるセッション番号を登録する処理は行わない。以下、新たに生成されたデータ・オブジェクトに対応するビュー・オブジェクトが生成されて、ビュー・オブジェクトがHTML文書を生成する。

【0051】ステップ1090で新たなデータ・オブジェクトを必要としない場合には、ビュー・オブジェクトが必要な子ビュー・オブジェクトをデータ・オブジェクト群及び端末属性情報を参照して生成し、ステップ1060(又はステップ1050)に移行する(ステップ1100)。すなわち、各ビュー・オブジェクトが端末属性情報を用いて担当部分のHTMLを出力し、クライアント端末に、生成したHTML文書を送信する。

【0052】では図5の例を用いてステップ1070以降の処理の一部を概説する。図9は、図5のHTML文書からユーザがABCを選択した場合を示す。HTTPのuseAgentフィールドに示されたdeviceIdは図5と同じように用いられる。一方、URLは例えば“http://aServer/servlet/RestaurantDemo?session=1&ABC=selected”となり、session=1の部分により、図5と同じセッション

```
<a href="/servlet/Restaurant?session=1&next=pressed">
  Next
</a>
```

【0056】アンカータグの中身は単なる文字列だけではなく、ボタンのイメージを指定することもできる。アンカータグの属性値として、パラメータ付きのURLを記述する。ユーザーがこのハイパーリンクをクリックした場合は端末からこのURLが送られ、本発明の制御装置にパラメータが渡され、session/パラメータを用いて特定されたビュー・オブジェクト群にnext=pressed/パラメータが渡される。このnext=pressedによりNEXTボタンが押されたことを検知して、ビュー・オブジェクトのHTML生成プログラムが次のページのHTML命令を出力する。

【0057】さらにユーザが図10のNEXTを指示した場合が図11に示されている。HTTPのUser-Agentフィールドの取扱いは同じである。URLは例えば“htt

であることが認識できる。さらに、ABC=selectedの部分から、図5のABCが選択されたことが分かるが、ビュー・オブジェクトでは新しいデータ・オブジェクトが必要であると判断される。そうすると、データ・サービス機構220は新たにデータ・オブジェクト群を生成する。そして、端末属性情報を用いて、ビュー・オブジェクト群を生成する。この例では、HTML文書を作成する際に、端末属性情報を参照して、ビュー・オブジェクト群のうち点線で囲まれた部分のみがHTMLを出力する。そして、生成されるHTML文書は一番下に表されたものである。ここでは、NEXTボタン(又はハイパーリンク)が生成されている。

【0053】ユーザが図9のNEXTを指示した場合が図10に示されている。HTTPのuseAgentフィールドの取扱いは同じである。URLは例えば“http://aServer/servlet/RestaurantDemo?session=1&next=pressed”となり、session=1の部分より図5と同じセッションであることが認識できる。さらに、next=pressedの部分から、図9のNEXTボタンが押されたことが分かる。しかし、ビュー・オブジェクトでは既存のビュー・オブジェクトで対応可能であると判断される。この場合には、データ・オブジェクト及びビュー・オブジェクトには何等の変更はなされず、ビュー・オブジェクト群の点線部分がHTMLを出力して、最終的なHTML文書を作成する。このHTML文書にはPREVボタン(又はハイパーリンク)及びNEXTボタン(又はハイパーリンク)が含まれる。

【0054】このような場合におけるNEXTボタン(あるいは前ページに戻るためのPREVボタン)のHTML出力例と動作例は次のようになる。

【0055】  
【表1】

p://aServer/servlet/RestaurantDemo?session=1&next=pressed”となり、session=1の部分より図5と同じセッションであることが認識できる。さらに、next=pressedの部分及びビュー・オブジェクトの内部状態から、図10のNEXTボタンが押されたことが分かる。ここで、ビュー・オブジェクト群は新たなビュー・オブジェクトが必要と判断する。そうした場合には、以前に生成したデータ・オブジェクト群及び端末属性情報を参照して、必要なビュー・オブジェクトを生成する。ここでは、ルート・ビュー・オブジェクト以外の点線部分が新たに生成される。そして、端末属性情報を参照しつつ、点線の部分のビュー・オブジェクト群がHTMLを生成し、HTML文書を作成する。

【0058】では、上では詳しく述べなかった端末属性

情報について述べる。図5乃至図11の例では、HTTPのUser-Agentフィールドに含まれるdeviceIDから端末属性情報を取得するとしているが、この対応情報が端末DB250に含まれている。例えば、deviceIDからその端末属性情報が得られるような対応テーブルを用意する。なお、端末属性情報には、図12に示されたようなものが考えられる。例えば、ScreenSize（画面サイズ）、ScreenRatio（縦横比）、ScreenQuality（画面の精細度）、ColorDepth（色数）、ModemSpeed（通信速度）、Image（イメージが使用可能か）、Table（テーブルは使用可能か）、Frame（フレームは使用可能か）、ScrollBar（スクロール可能か）、PointingDevice（ポインティング・デバイスは使用可能か）等がある。図12の右列にはそれぞれの値の例を示している。なお、全てのdeviceIDについて全ての項目が指定されている必要はない。また、各行の値はその他の値をとるようにしてもよい。

【0059】さらに、もう一段階処理を行うことも考えられる。すなわち、端末属性ルールベースを用意し、図12のような端末の特性値から、表示構造の変換や、表示命令の生成に使用するより一般的な端末属性を生成するようにする。例えば、ModemSpeedが14.4Kより小さい場合には通信速度が遅い、ModemSpeedが56Kより大きければ通信速度が速い、それ以外を中程度の速度といったように、単一パラメータを具体的な数値でなく範囲を決めて分類したり、複数のパラメータから例えばイメージは表示速度が遅い、ポインティング・デバイスがなく通信速度が遅いので次元表示にする等の端末属性を生成することも考えられる。

【0060】また、上ではセッションの開始のみを説明したが、ユーザがセッションの終了を明示的に示した時にはそれにより、明示的なセッションの終了がない場合には直前のアクセスから所定の時間経過後に、セッションの終了とする。セッションが終了すると、それまでデータ・オブジェクト群及びビュー・オブジェクト群により使用されてきたリソースは解放される。

【0061】ここでデータ・オブジェクトの生成方法について簡単に説明しておく。Webアプリケーションのコンテンツ作成者はアプリケーション定義文書300

（図13）を作成する。アプリケーション定義文書300はある種の定義言語にしたがって記述される。この言語は、例えば、XML（W3C, "Extensible Markup Language (XML) 1.0", <http://www.w3.org/TR/1998/REC-xml-19980210>）を用いて独自のマークアップ言語として定義することができる。レストランサービスの例では図14のような記述例が考えられる。

【0062】ここでは、今定義しようとしているレストラン情報サービスが、レストラン検索とレストラン情報表示という2つのプロセス（表示上のページに相当する）からなり、さらにレストラン情報表示プロセスは、

レストラン情報という表示項目とレストラン検索のページに戻るためのリンクからなる。同様に、レストラン情報項目はその下のレベルの表示項目から構成されることが記述される。図14では、部分的に省略が行われている。このような定義文書の作成は、テキスト・エディタを用いてコンテンツ作成者が直接記述してもよいし、レイアウト機能を持つワープロのようなオーサリング・ツールを用いてもよい。

【0063】図14のアプリケーション定義言語に再実行制御の指定する例を、さらに、図15を参照して説明する。図14のアプリケーション定義言語に、さらに、図15に例示するように、レストランの予約処理のプロセスが追加される。この予約処理の中には、大きく、予約状況を表示する表示項目と予約申し込みを入力させて予約を受け付ける表示項目がある。これらの表示項目は、属性として「予約受付後再処理禁止」というページグループに属することが記述されている（pageGrp=のキーワード）。さらに、予約処理を実行する決定ボタンの属性として、予約が受け付けられた場合、このページグループに属する表示項目は再処理されないことが指定されている（invGrp=のキーワード）。

【0064】したがって、この記述にしたがって生成された表示制御情報がブラウザで処理された画面に対して、ユーザが決定ボタンを押した後は、本発明の機構により、これらの表示項目およびそれより下のレベルの表示項目は、再処理（表示や再予約の受付）は禁止される。予約が受付られた後に、ユーザがブラウザのバックボタンで「予約状況表示」の表示項目（ページ）に戻って、再表示しようとしても2重予約防止のために表示は行われない。

【0065】ページの再実行制御は、動的に生成される画面にも対応するため、ページ定義ツリーを使って、ページグループを定義する。上記の図16は図15のレストランの例で、特に予約処理プロセスの部分のページ定義ツリーを示したものである。予約処理プロセスは、予約状況表示、予約受付画面、予約受付結果画面、の3つの表示項目を持つ。さらに、予約状況表示は、予約日時、その空き情報の2種類のサブ表示項目をもつ。予約受付画面についても同様である。

【0066】これらの表示項目は、端末の画面表示能力により、動的にページとして表示され、どのレベルがページになるかはあらかじめ決められていない。例えば、PCのような画面サイズが大きいものは、予約状況表示や予約受付画面がページの単位となるが、より小さい画面のPDAのようなものなら、個々のサブ表示項目である予約日時と空き情報で1ページになるかもしれない。

【0067】ここで、「予約受付後再処理禁止」というページグループは、図15のアプリケーション定義言語では、予約状況表示と予約受付画面の2つの表示項目の定義に指定されているだけであるが、ページ定義ツリー

の子供もそのページグループに属する。すなわち、「予約受付後再処理禁止」というページグループは、上記の図の点線内の表示項目（すなわちページ）をメンバーとして持つ。

【0068】再び図13を参照する。定義言語解析器310は、このような言語で書かれた文書を解析するものである。このような解析器の例としては、IBM XML for Java (<http://www.alphaworks.ibm.com/formula/xml>)がある。解析器は解析結果をデータ・オブジェクト生成器に送る。解析結果はW3C, "Document Object Model Specification" (<http://www.w3.org/TR/WD-DOM-1998-0416>)に述べられているように、タグの構造を一つのオブジェクトにしたようなオブジェクトの木構造（入れ子の内部にあるタグのオブジェクトが子どもとなる）で表現されるデータである。

【0069】データ・オブジェクト生成器は、解析結果の木構造を元にデータ・オブジェクトを生成する。一般的には解析結果のオブジェクトからデータ・オブジェクトのクラスを決定しオブジェクトを生成する。構造も同じ形になる。

【0070】アプリケーション定義文書にすべての情報を書いておくかわりに、情報の内容を既存のデータベースから取ってくることもできる。この場合、定義文書にはデータベースの検索方法の指定や検索結果の参照が記述され、データ・オブジェクト生成器がこの指示を元にデータベース・コネクタというデータベース・システムとのやり取りを仲介機構としてデータベースに問い合わせを行う。データベースの検索結果がデータベース・コネクタを通して返されるので、データ・オブジェクト生成器は検索結果の参照指示を元にデータベースからの結果を反映したデータ・オブジェクトを生成する。このときの構造は定義文書のデータベースへの指示が行われているタグによって決定される。このようなデータベースの指示を含む定義文書の例（図17）として、図14のレストラン情報をデータベースから動的に取ってくる場合を挙げる。

【0071】このような処理を実施することにより生成されるデータ・オブジェクト群の例を図18に示す。ここで出力を要求されている「レストラン詳細ページ」は「レストラン情報」と検索画面に戻るための「サーチページへのリンク」の2つのオブジェクトから構成される。さらに、レストラン情報は、ユーザの欲するレストランに関する様々な情報を保持するが、ここでは「属性」「ガイド」「メニュー」の3つのカテゴリに分けられる。各カテゴリの下には必要な情報に対応するオブジェクトが生成されている。

【0072】本発明では、データ・オブジェクトに、図19に示すような属性が定義されている。子オブジェクトが存在している場合には子オブジェクトをその集合として、データの優先度を数値化したものを番号で、その

オブジェクトのデータを分割しても常に表示するか否かをスイッチ（論理値）で、このオブジェクトのデータを分割表示できるか否かをスイッチで、イメージの場合にはアウトライン化できるかどうかをスイッチで、その他アプリケーションに固有のデータを必要に応じて任意の型で定義する。これが、対応ビュー・オブジェクトの生成及びビュー・オブジェクトによる前記表示制御情報の生成において用いられる指示情報である。

【0073】このデータ・オブジェクトが生成された場合に、どのようなビュー・オブジェクトが生成され、どのようなHTML文書が生成されるかについて、以下に2つの代表的な例を示す。

【0074】（1）PC等の大画面用の表示構成への変換とHTML出力

先に述べた例では、クライアント端末から受信するHTTPのUser-Agentフィールドから、deviceIdを取り出し、対応する端末属性情報を取得する。以下の説明では、この端末属性情報を既に取得しているものとして説明する。本例では、端末属性情報は、「大画面」（より具体的には、大画面と分類されるような解像度の表示装置をクライアント端末が有している。）という情報を含んでいるものとする。

【0075】まず、データ・オブジェクトの木構造のルートとなるオブジェクトに対応するビュー・オブジェクトを生成する（図20）。これを行うHTML生成機構230は、ルート・データ・オブジェクトのクラス（例では「ページクラス」）や、アプリケーション分野の固有知識（例えば「1つのレストランは1ページで表示する」）、そして、端末属性情報（ここでは「大画面」）をパラメータとして、ビュー・オブジェクトのクラス（大画面用ページ表示ビュー・オブジェクトのクラス）とオブジェクトを生成するときの初期パラメータを決定し、実際にルート・ビュー・オブジェクトを生成する。

【0076】このようにして、ひとたびルート・データ・オブジェクトに対応するルート・ビュー・オブジェクトが決定されると、このルート・ビュー・オブジェクトを出発点として、ビュー・オブジェクトに自分の子供を次々に生成させていくことができる。すなわち、ビュー・オブジェクトは子オブジェクトを生成するメソッド又は共通の子ビュー・オブジェクトを生成するメソッドに子ビュー・オブジェクトを生成するように依頼するメソッドを有している。

【0077】そして、大画面用ページ表示ビュー・オブジェクトは子ビュー・オブジェクトの生成を依頼されると、自分自身に対応するデータ・オブジェクトである「レストラン詳細表示ページ・データ・オブジェクト」に問い合わせ、その子データ・オブジェクトから自分の子ビュー・オブジェクトを生成する。この生成において、一般的には、自分自身のクラス（ここでは「大画面用ページ表示ビュー・クラス」）、データ・オブジェク

トのクラス、アプリケーション分野の固有知識（定義されていれば）、及び端末属性情報から決定される、ビュー・オブジェクトのクラスと、初期パラメータとを元にオブジェクトの生成が行われる。

【0078】例えば、図21に示すように、「レストラン詳細表示ページ」データ・オブジェクトの子である「レストラン情報」の場合は、「レストラン情報」のクラスである「一般的項目」と端末属性情報の「大画面」情報とから、ビュー・オブジェクトのクラスとして「項目2次元配置表示」クラスを決定している。この項目2次元配置表示ビュー・オブジェクトは、端末画面が広いことを利用して出力対象を2次元的に配置して表示効果を高めることを意図したビュー・オブジェクトの一種である。同様に「レストラン詳細表示ページ」データ・オブジェクトのもう一つの子オブジェクトの「サーチページへのリンク」の場合は、そのクラス（「ハイパーリンク」）から「リンク・オブジェクト」というビュー・オブジェクトを生成している。

【0079】子ビュー・オブジェクトが生成されたら、親ビュー・オブジェクトはすべての子に対し、自分が依頼されたのと同様な子ビュー・オブジェクトの生成依頼を行う。すなわち、図22に示すように、自分自身が2次元配置ビュー・オブジェクトであること、「レストラン情報」の子データ・オブジェクトの「属性情報カテゴリ」が属性の一覧を表示することなどを利用して、「属性一覧表示」ビュー・オブジェクトを生成する。同様にデータ・オブジェクトである「ガイド情報カテゴリ」データ・オブジェクトに対して、ユーザの要求や端末属性情報から最適の表示項目を一つ選ぶという振る舞いをする「属性選択表示」ビュー・オブジェクトが生成される。「メニュー」データ・オブジェクトに対してはレストラン情報と同様に端末属性情報の「大画面」を利用して、再び項目2次元配置ビュー・オブジェクトが生成される。更に、レストラン情報ではレストラン名がタイトルとして重要であるという分野固有の情報を利用して、レストラン名をタイトル表示するビュー・オブジェクトが子オブジェクトとして加えられている。

【0080】これら、4つの子ビュー・オブジェクト「レストラン名表示」、「属性一覧表示」、「属性選択表示」、「メニューの項目2次元配置」に対して、同様に子ビュー・オブジェクトの生成を依頼していく。例えば、「属性一覧表示」ビュー・オブジェクトが生成を依頼されると、対応する「属性情報カテゴリ」の子データ・オブジェクトを走査し、変数名と値のペアのビュー・オブジェクト（属性名値ペア表示ビュー・オブジェクト）を生成する。「属性選択表示」ビュー・オブジェクトでは、「ガイド情報カテゴリ」データ・オブジェクトの子から最も適切なものを1つ選び（ここでは、地図機能がないPCで、データ・オブジェクトの優先度から、イメージの案内図が選択されるとする）、それに対応す

るビュー・オブジェクトを生成する。メニューに関しても同様に再帰的処理が行われる。最終的に生成されるビュー・オブジェクト群の構造は図23のようになる。

【0081】ここでデータ・オブジェクト群の構造を示す図18とビュー・オブジェクト群の構造を示す図23を比較すると、レストラン名ビュー・オブジェクトが追加されており、「ガイド情報カテゴリ」データ・オブジェクトの下に4つの子データ・オブジェクトが存在しているが、属性選択表示ビュー・オブジェクトが「ガイド情報カテゴリ」データ・オブジェクトに対して生成されているので、属性選択表示ビュー・オブジェクトの子ビュー・オブジェクトは1つしかない。それから、メニュー名ビュー・オブジェクトが重要性から追加されている。以上のように、データ・オブジェクトとビュー・オブジェクトは構造が必ずしも1対1にはならない。

【0082】すべてのビュー・オブジェクトが生成されると、HTMLの生成が各ビュー・オブジェクトに依頼される。すなわち、ルート・ビュー・オブジェクトにHTML生成が指示され、大画面用ページ表示ビュー・オブジェクトは最終出力中のHTMLページの制御命令を生成するがページの中身に関しては子である「項目2次元配置ビュー・オブジェクト」や「ハイパーリンク」オブジェクトに依頼をしていく。依頼された子ビュー・オブジェクトは更に自分の子ビュー・オブジェクトに依頼をしながら全体のHTMLを生成していく。

【0083】図24に図23のビュー・オブジェクト群によるHTML生成の概要を示す。大画面用ページ表示ビュー・オブジェクトの子である項目2次元配置表示ビュー・オブジェクトは最終的に2次元に配置されるようにHTMLを生成していく。以下で説明するように、このHTMLを生成する段階でさらに端末属性情報を参照してHTML出力を変えていくことができる。図24では、項目2次元配置表示ビュー・オブジェクトの子ビュー・オブジェクトは、図示のとおり、2次元配置するようにHTMLを出力している。なお、属性一覧表示ビュー・オブジェクトの子ビュー・オブジェクト、2つの項目2次元配置表示ビュー・オブジェクトの子ビュー・オブジェクトについては図面上は表示されていないが、各親ビュー・オブジェクトにおいて確保された領域内に表示されるようにHTMLを出力している。

【0084】一方、属性選択表示ビュー・オブジェクトの子ビュー・オブジェクトであるイメージ表示ビュー・オブジェクトについては、イメージ表示速度が十分に速いと解釈できるような端末属性情報がある場合には、イメージをそのまま出力するようにする（図25参照）。イメージ表示速度が十分に速いという端末属性は、通信速度及び描画処理能力により決定されるが、単に通信速度が遅いことで以上の処理を実施するようにしてもよい。

【0085】他方、イメージ表示速度が遅いと解釈でき

るような端末属性情報がある場合、イメージ表示ビュー・オブジェクトは、イメージをそのまま出力することなく、イメージへのリンクを出力する(図26参照)。ユーザがこのイメージを指定した場合には、このイメージを送信することにする。

【0086】また、2次元配置の方法にしても、画面サイズや縦横比などを考慮して、クライアント端末ごとに調整することが考えられる。このようにして、「端末の画面が大きい」「イメージの表示速度が遅い」といった端末属性情報に適合させてHTMLを生成していくことができる。

【0087】(2)PDAなどの小画面向けの表示への変換とHTML出力

PDA等の小画面では、一時に表示される情報の量に制限があるので、ビュー・オブジェクトの構成とその木構造をそれに適したものにしていなければならない。端末属性情報は、小画面と解釈されるようなデータを含んでいるとする。

【0088】データ・オブジェクトは図18であるとする。まず、大画面のときと同様にデータ・オブジェクトのルート・オブジェクトのクラス、端末属性情報(この場合は「画面が小さい」)を用いて、HTML生成機構230が小画面用ページ分割表示ビュー・オブジェクトを生成する(図27参照)。ルート・ビュー・オブジェクトが生成されたので、大画面のときと同じように子ビュー・オブジェクトの生成を行う。

【0089】ただし、今回は一画面に表示できる情報量が限られること、情報が一度に提示されずに分割されることから、2次元配置を行うようなビュー・オブジェクトではなく、情報の重要度が高いものから順番に表示するようなビュー・オブジェクトが生成される。このビュー・オブジェクトが更に子オブジェクトを生成するときには、2次元配置に向けたカテゴリ構造ではなく、表示内容の重要度を考慮した列構造となるようにビュー・オブジェクトを生成する。

【0090】よって図28に示すように、レストラン情報データ・オブジェクトに対応して、優先度を考慮した線形表示ビュー・オブジェクトを生成する。リンク・オブジェクトを生成するのは図21と同じである。

【0091】そして、優先度を考慮した線形表示ビュー・オブジェクトが子ビュー・オブジェクトを生成する際には、ユーザにいち早く見せるべきものは何かということ、及びページ分割しやすい様に列構造にすることが重視され、クライアント端末が大画面を有するときとはまったく異なるビュー・オブジェクトの木構造となる(図29参照)。本例では、その時点で実際に表示されるビュー・オブジェクト以外にも生成しているが、ユーザにより指示された際に生成するようにすることも可能である。図29では、優先度を考慮した線形表示ビュー・オブジェクトに、必要な子ビュー・オブジェクトが並列に

接続される形になる。図では優先度を考慮した線形表示ビュー・オブジェクトに近いほど優先度が高くなっている。なお、何等かの方法でユーザの好みを反映してこの優先度を決定することも考えられる。

【0092】ビュー・オブジェクトが生成された後は、大画面のときと同様にHTMLの生成をルート・ビュー・オブジェクトから順に行う。分割されたページをユーザが容易にナビゲートできるようにページ間のナビゲート・ボタン(NEXTやPREVボタン)も自動生成される。このとき、端末属性情報を使って、端末の能力に、より適合させた出力を生成させることができる。

【0093】図30の場合には、端末属性情報に含まれる「画面サイズ320×240dot」という情報から優先度の高い順に3つのビュー・オブジェクトの情報を1ページで表示するようにHTMLを生成させることができる。さらに、「画面サイズ240×180dot」という端末属性情報があれば、さらに画面が小さいので、2つのビュー・オブジェクトの情報を1ページで出力するようにHTMLを生成させることもできる。

【0094】以上の図では、HTMLをWebブラウザが解釈した後の画面をビュー・オブジェクト群の出力として表しているが、ビュー・オブジェクト群はHTMLの制御命令を出力するものである。

【0095】以上の例ではHTMLを対象に説明してきたが、本発明はこれに限定されるものではない。例えば、WML(Wireless Markup Language, Wireless Application Protocol Forum Ltd, "Wireless Application Protocol Wireless Markup Language Specification", Version 30-Apr-1998)やXML等の出力を必要とするならば、それらを出力するようなメソッドを含むビュー・オブジェクトを生成するようにし、且つWMLやXMLの特性に合うようなビュー・オブジェクト群の構造を生成するようにすればよい。

【0096】上記の図31は、処理に関連してページの再実行の制御を行う機構を説明するために、本発明の動作を時間をおって、レストランの予約処理を例に説明した図である。同時に、端末の表示画面、ユーザの操作(サーバーへの要求内容)、ページ管理DB 213内の処理状態キャッシュの内容もあわせて表示している。まず、ユーザからレストランの予約状況表示要求が発せられる。これにより、本発明の制御機構210が動作し、その中のページ管理機構212により新しいページであることが判別される。次にアプリケーション処理機構214が実行される。ここでは、予約状況がアプリケーションプログラムによって調べられる。同時に、再実行に備えてこのときの処理状態が状態キャッシュに格納される。この処理では再実行制御は必要ないので、次の処理である画面作成が行われる。この結果、予約処理状況を表示するための表示制御情報が端末に送られ、表示される。予約受付画面も同様に処理され、状態キャッシュに

状態が格納される。

【0097】次に、ユーザが端末上の履歴機能を使って、予約状況画面を再表示させようとしたとする。例えば、ブラウザに通常備わっているバックボタンで古い予約状況画面を呼び出し、同じく通常備わっている再表示ボタンを押したとする。この場合、ページ管理機構 212 が古い画面に対する処理であることを判別する。この場合、履歴上のページであるので、状態キャッシュに状態が格納されているかをチェックする。この場合は、上記に述べたように状態キャッシュに格納されているはずなので、その処理状態を復元しアプリケーションの処理（予約状況調査）を行う。正確には、この結果も状態キャッシュに格納されるが、図では簡単化のために省いてある。最後に表示制御情報が生成され端末に送られる。

【0098】引き続き、レストランの予約処理の例で、再実行が禁止される様子を上記の図32により説明する。予約受付画面にユーザが必要事項を入力し、予約決定ボタンを押したとする。これにより、本発明の制御機構 210 が動作し、その中のページ管理機構 212 により新しいページであることが判別される。次にアプリケーション処理機構 214 が実行される。ここでは、予約受付処理がアプリケーションプログラムによって実行される。同時に、再実行に備えてこのときの処理状態が状態キャッシュに格納される。この処理では予約受付により予約関係の再実行制御が指定されているので、アプリケーション定義言語内のinvGrpの指定「予約受付後再実行禁止」をもとに再実行が禁止されるページが同定される。この例の場合、アプリケーション記述言語の表示項目属性 pageGrpが指定されている、「予約状況表示」「予約受付画面」のページが該当する。次に、これらのページに相当する状態キャッシュが無効化され、受付結果の画面作成が行われる。

【0099】次に、前と同様に、ユーザが端末上の履歴機能を使って、予約状況画面を再表示させようとしたとする。例えば、ブラウザに通常備わっているバックボタンで古い予約状況画面を呼び出し、同じく通常備わっている再表示ボタンを押したとする。この場合、ページ管理機構 212 が古い画面に対する処理であることを判別する。この場合、履歴上のページであるので、状態キャッシュに状態が格納されているかをチェックする。前の場合と異なり、上記に述べたように状態キャッシュは無効化されているので、再実行が禁止されていることが判明する。この結果、エラーを通知する表示制御情報が生成され端末に送られて、エラー画面が端末に表示される。

【0100】さらに、これまでのレストランの予約処理の例で、画面構成が（端末属性により）動的に変化する場合でも再実行制御の機構が問題なく動作することを、上記の図33により説明する。これまでの例では、画面が大きい端末であったので、予約状況表示は1つの画面

で表示されていた。今度は、画面の小さい端末を想定し、HTML生成機構 230 が予約状況表示の子供のオブジェクト「予約日時」「空き情報」の項目ごとにページを分割する。したがって、状態キャッシュもこれらの子供の項目ごとにキャッシュされることになる。ここで、前と同様に、予約受付画面にユーザが必要事項を入力し、予約決定ボタンを押したとする。

【0101】これにより、本発明の制御機構 210 が動作し、その中のページ管理機構 212により新しいページであることが判別される。次にアプリケーション処理機構 214 が実行される。ここでは、予約受付処理がアプリケーションプログラムによって実行される。同時に、再実行に備えてこのときの処理状態が状態キャッシュに格納される。この処理では予約受付により予約関係の再実行制御が指定されているので、アプリケーション定義言語内のinvGrpの指定「予約受付後再実行禁止」をもとに再実行が禁止されるページが同定される。この例の場合、アプリケーション記述言語の表示項目属性 pageGrpが指定されている、「予約状況表示」「予約受付画面」のページが該当する。しかし、予約状況表示のページは直接にはキャッシュ内に存在しないので、ページ定義ツリーにしたがって、予約状況表示の子供である「予約日時」と「空き情報」が該当するページであることを用いて、これらのページに相当する状態キャッシュが無効化される。

【0102】次に、前と同様に、ユーザが端末上の履歴機能を使って、予約状況画面のひとつである予約日時と空き情報の1ページを再表示させようとしたとする。例えば、ブラウザに通常備わっているバックボタンで古い予約状況画面を呼び出し、同じく通常備わっている再表示ボタンを押したとする。この場合、ページ管理機構 212 が古い画面に対する処理であることを判別する。この場合、履歴上のページであるので、状態キャッシュに状態が格納されているかをチェックする。前の場合と同様に、上記に述べたように状態キャッシュは無効化されているので、再実行が禁止されていることが判明する。この結果、エラーを通知する表示制御情報が生成され端末に送られて、エラー画面が端末に表示される。

【0103】実際に表示されるページを直接指定するのではなく、ページグループという単位で指定するため、実際に表示されるページが分割されても、アプリケーションで要請される正しい再実行制御をすることができた。

【0104】

【効果】様々な能力を有するクライアント端末に対して、それぞれの能力に応じたユーザ・インターフェースを提供するための動的な文書又は表示制御情報を生成することができた。

【0105】また、サーバで実施されるアプリケーションを、当該アプリケーションの処理結果を使用するクラ

クライアント端末に依存しないようにする機構を提供することもできた。

【0106】さらに、サーバで実施されるアプリケーションの処理と当該処理結果を表す表示文書の作成とを分離し、様々なクライアント端末が出現しても、表示文書の作成部分のみを変更すればよいようにすることもできた。

【0107】さらに、端末のブラウザの画面再表示機能をもちいた履歴上のページの再実行に対して、アプリケーションの処理にあわせた柔軟な再実行制御と認証制御を、アプリケーションプログラムからの簡単な指定で実現することができた。この指定は、アプリケーション処理の記述にページのグループとして簡単に記述し、動的なページ構成を行うときにも有効であることも示せた。

【図面の簡単な説明】

【図1】本発明のシステム全体を表す図である。

【図2】HTTPサーバ100内の機能ブロック図である。

【図3】図2に示した制御機構の機能ブロック図である。

【図4】HTTPサーバ100の処理フローを説明するための図である。

【図5】処理概要の一例を示した図である。

【図6】図4に示したステップ1070の処理フローを説明するための図である。

【図7】図6に示したステップ1072の処理フローを説明するための図である。

【図8】図6に示したステップ1075の処理フローを説明するための図である。

【図9】処理概要の一例を示した図である。

【図10】処理概要の一例を示した図である。

【図11】処理概要の一例を示した図である。

【図12】端末属性情報の内容の一例を表す図である。

【図13】データ・オブジェクトを生成する際に用いられる機能ブロック図である。

【図14】アプリケーション定義文書の一例を表す図である。

【図15】図14のアプリケーション定義言語に再実行制御の指定する例を示す図である。

【図16】図15のレストランの例で、特に予約処理プロセスの部分のページ定義ツリーを示したものである。

【図17】データベースの指示を含む定義文書の一例を示す図である。

【図18】データ・オブジェクト群の一例を示す図であ

る。

【図19】アプリケーション・オブジェクトに定義される属性の一例を示す図である。

【図20】大画面用のビュー・オブジェクトを生成する際の処理を説明する図である。

【図21】大画面用のビュー・オブジェクトを生成する際の処理を説明する図である。

【図22】大画面用のビュー・オブジェクトを生成する際の処理を説明する図である。

【図23】大画面用のビュー・オブジェクト群を表す図である。

【図24】図23のビュー・オブジェクト群がHTMLを出力する際の処理を説明するための図である。

【図25】図23のビュー・オブジェクト群がHTMLを出力する際の処理を説明するための図である。

【図26】図23のビュー・オブジェクト群がHTMLを出力する際の処理を説明するための図である。

【図27】小画面用のビュー・オブジェクトを生成する際の処理を説明する図である。

【図28】小画面用のビュー・オブジェクトを生成する際の処理を説明する図である。

【図29】小画面用のビュー・オブジェクト群を表す図である。

【図30】図23のビュー・オブジェクト群がHTMLを出力する際の処理を説明するための図である。

【図31】処理に関連してページの再実行の制御を行う機構を説明するために、本発明の動作を時間をおって、レストランの予約処理を例に説明した図である。

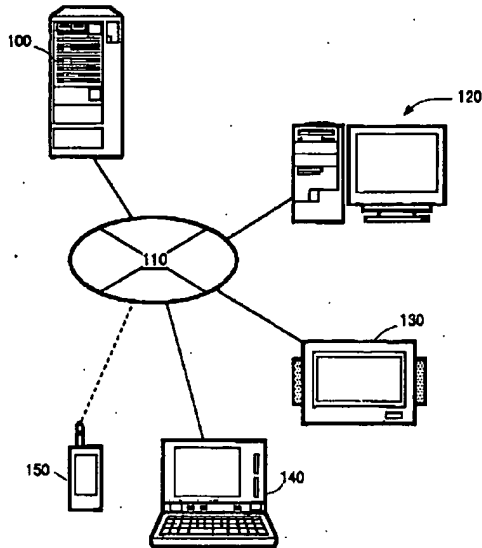
【図32】レストランの予約処理の例で、再実行が禁止される様子を説明する図である。

【図33】これまでのレストランの予約処理の例で、画面構成が（端末属性により）動的に変化する場合でも再実行制御の機構が問題なく動作することを説明する図である。

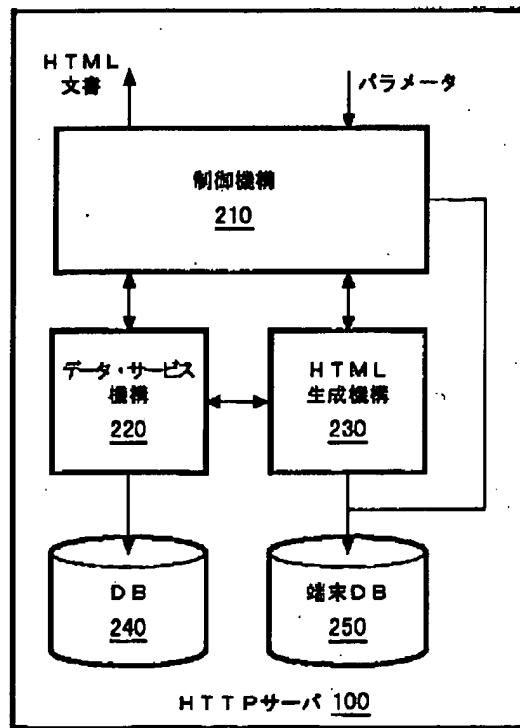
【符号の説明】

100 HTTPサーバ  
110 ネットワーク  
120-150 クライアント端末  
210 制御機構  
220 データ・サービス機構  
230 HTML生成機構  
240 DB  
250 端末DB

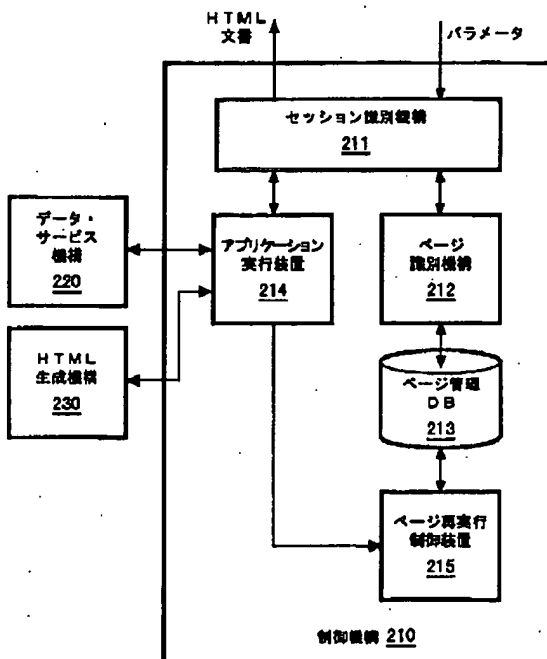
【図1】



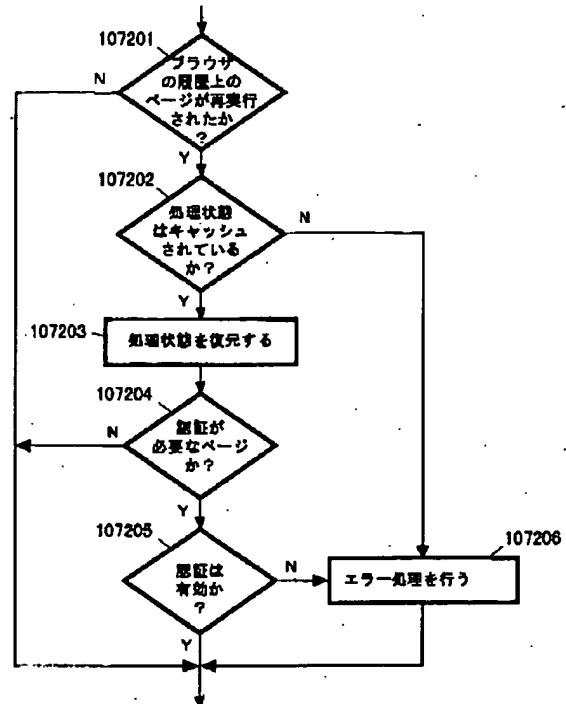
【図2】



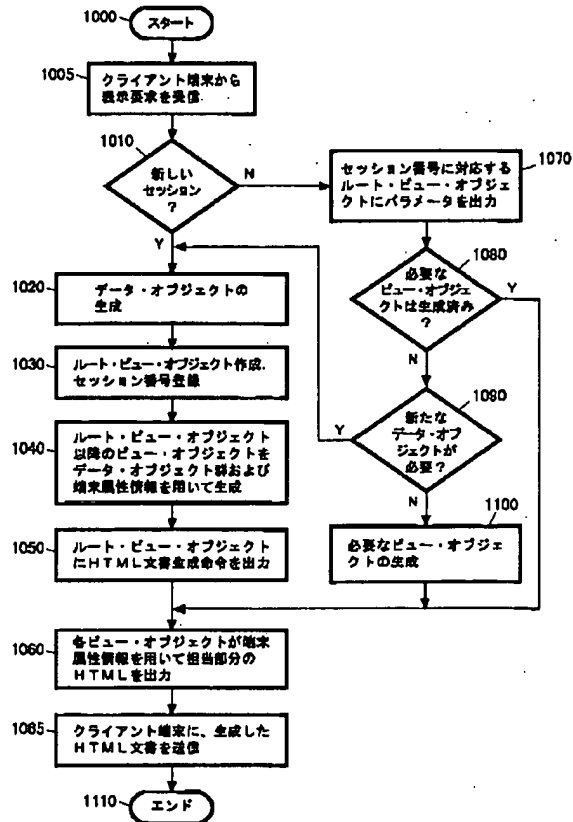
【図3】



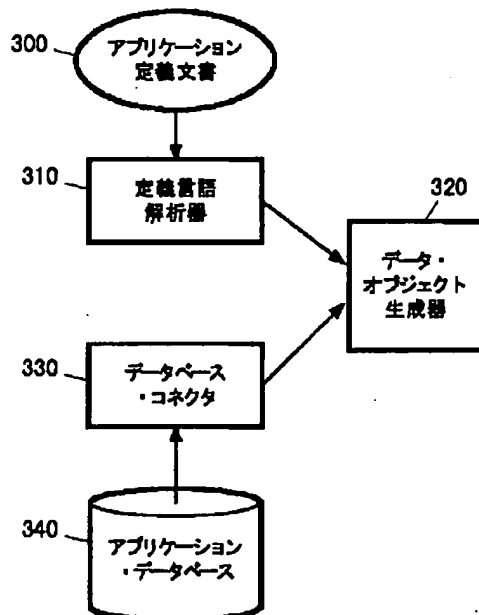
【図7】



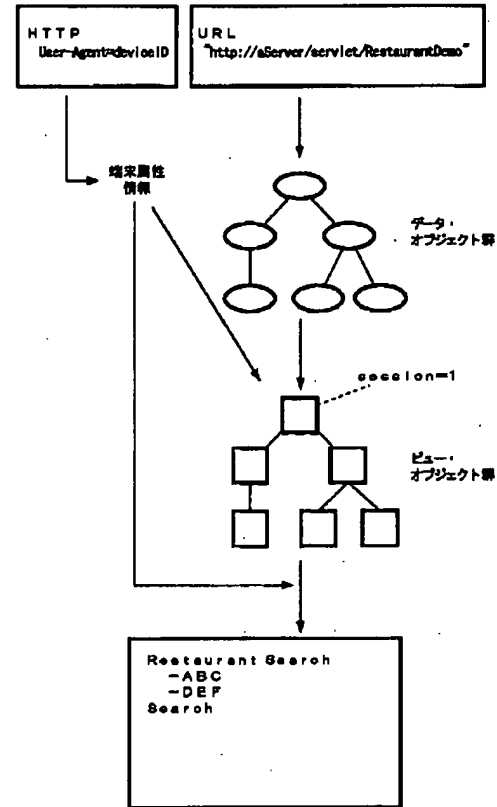
【図4】



【図13】



【図5】

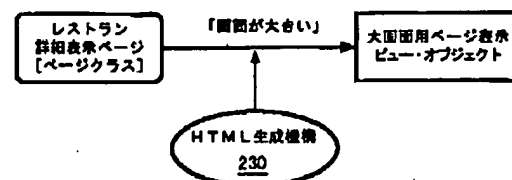


【図14】

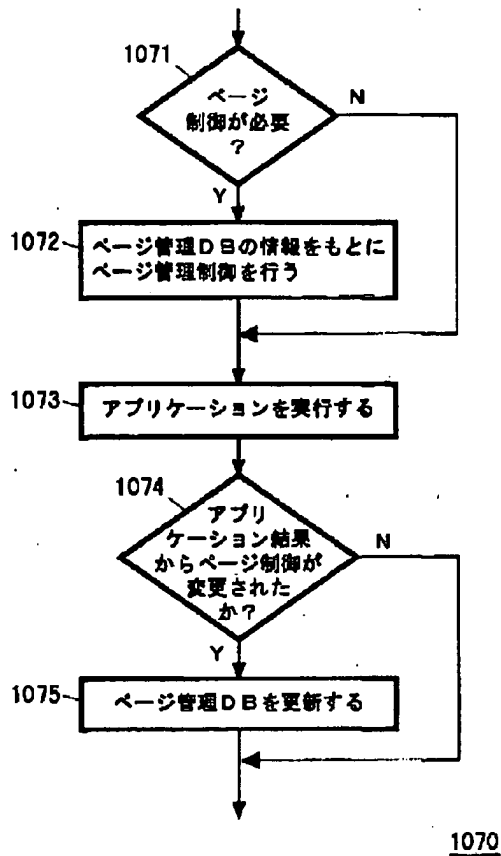
```

<Webサービス name="レストラン情報サービス">
  <プロセス name="レストラン検索">
    ...
  </プロセス>
  <プロセス name="レストラン情報表示">
    <表示項目 name="レストラン情報">
      <表示項目 name="レストラン属性一覧">
        <表示項目 name="ジャンル">和食</表示項目>
        <表示項目 name="開店時間">11:00-23:00</表示項目>
      </表示項目>
      <表示項目 name="ガイド情報">
        ...
      </表示項目>
    </表示項目>
  </プロセス>
</Webサービス>
  
```

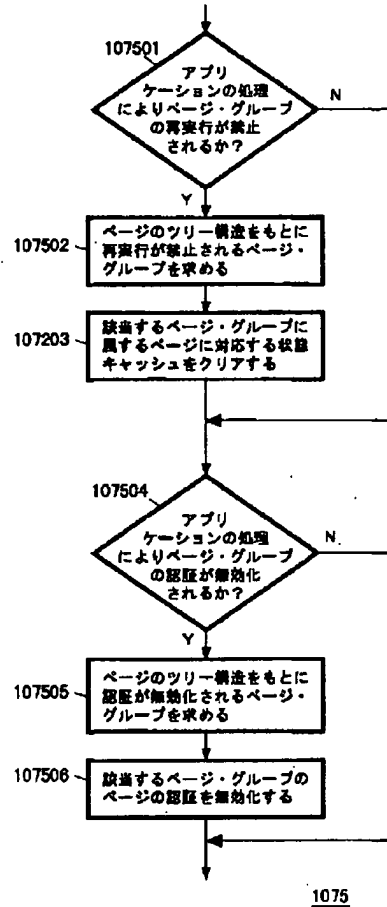
【図20】



【図6】



【図8】



【図15】

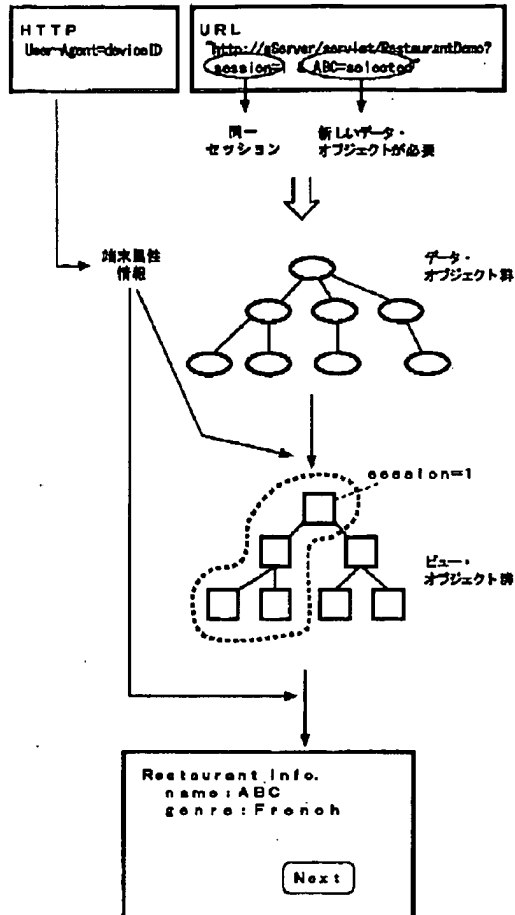
```

</プロセス name="予約処理">
  <表示項目 name="予約状況表示">
    pageGrp="予約受付後再処理禁止"
    <表示項目 name="予約日時"> 3/1 13:00 ...
    <表示項目 name="空き情報"> 3テーブル空き ...
    <表示項目 name="予約日時"> 3/2 18:00 ...
    <表示項目 name="空き情報"> 10テーブル空き ...
  </表示項目>
  <表示項目 name="予約受付画面">
    pageGrp="予約受付後再処理禁止"
  <入力項目 name="希望日時">
  <決定ボタン name="予約">
    invGrp="予約受付後再処理禁止"
  </表示項目>
  <表示項目 name="予約受付結果画面"> ...
</プロセス>
  
```

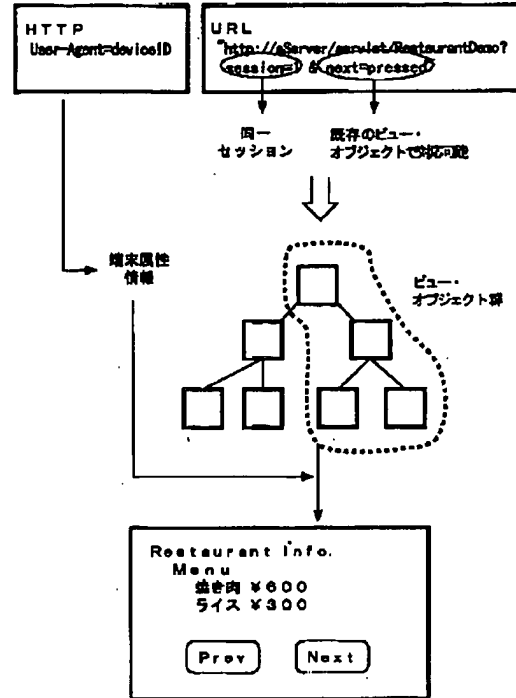
【図19】

属性名	型	説明
子オブジェクト	データ・オブジェクトの集合	本構造を定義する
優先度	番号	データの重要度を数値化
常に表示	スイッチ	分割しても常に表示するかどうか
分割不可	スイッチ	このオブジェクト内容を分割してはいけないかどうか
イメージのアウトライン化	スイッチ	このイメージはアウトライン化してもいいかどうか
(アプリケーション固有の属性データ)	(任意の型)	個々のアプリケーションに特有のデータを定義する

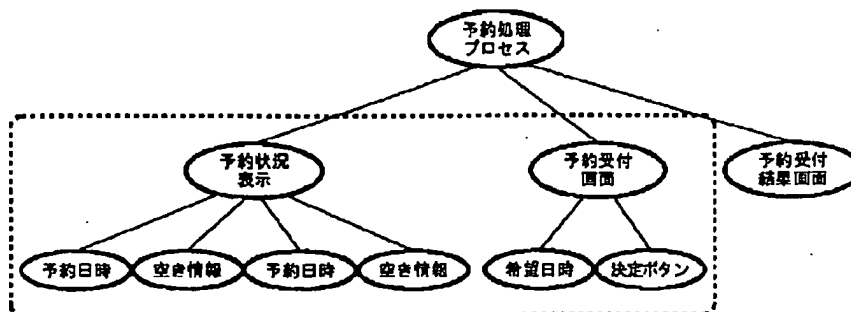
【図9】



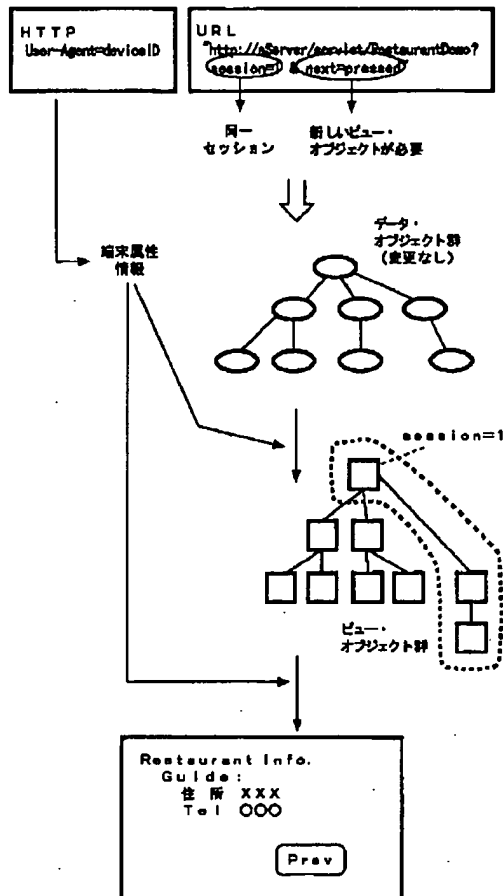
【図10】



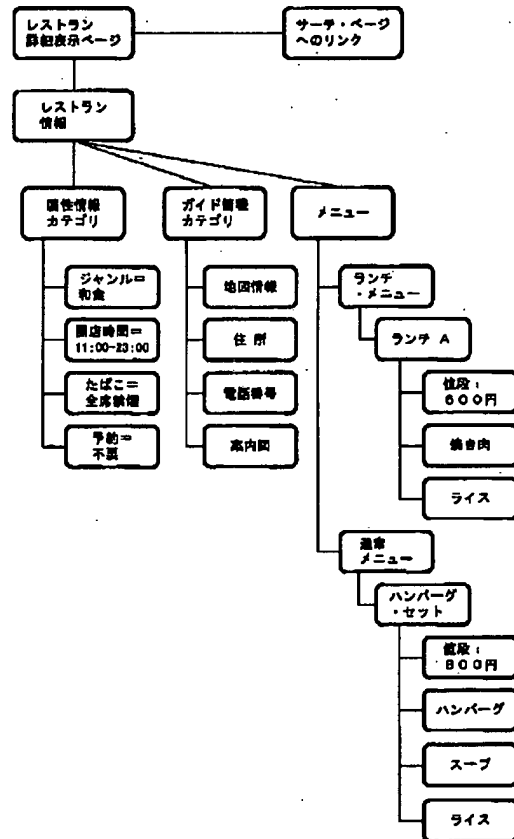
【図16】



【図11】



【図18】



【図17】

```

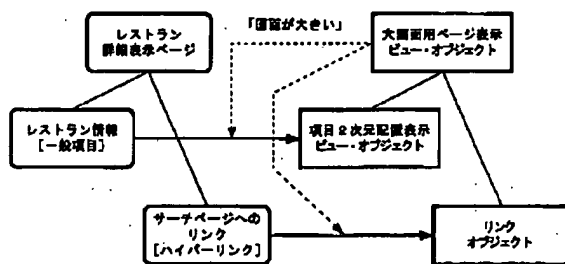
<Webサービス name="レストラン情報サービス">
  <プロセス name="レストラン検索">
    .
    .
  </プロセス>
  <プロセス name="レストラン情報表示">
    <DB参照表示項目 name="レストラン情報" SQL="select * from XXX"/>
    <DB参照表示項目 name="レストラン属性一覧">
      <DB参照表示項目 name="ジャンル" dbref="genre"/>
      <DB参照表示項目 name="開店時間" dbref="open"/>
      .
    </DB参照表示項目>
    <DB参照表示項目 name="ガイド情報">
      .
    </DB参照表示項目>
    <リンク process="レストラン検索"/>
  </プロセス>
</Webサービス>

```

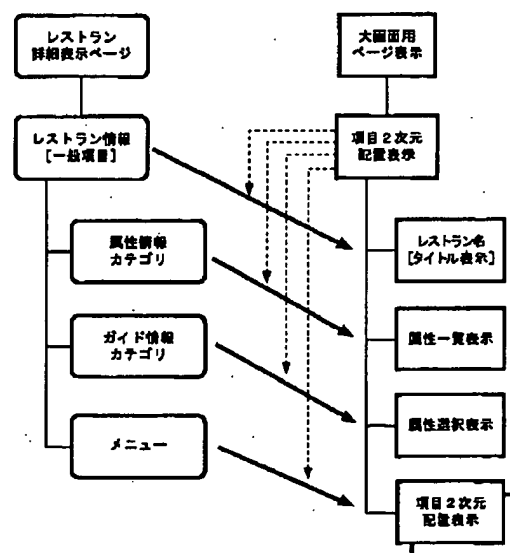
【図12】

変数名	値の例
ScreenSize (画面サイズ)	640×480, 1280×768, 1600×1600
ScreenRatio (縦横比)	4:3, 16:9, 1:1, 3:4, 9:16
ScreenQuality (画面の精細度)	excellent, good, normal, bad
ColorDepth (色数)	2(白、黒), 16, 256, 16M
ModemSpeed (通信速度)	14.4K, 33.6K, 56K, 128K, unlimit
Image (イメージは使用可能か)	yes, slow, no
Table (テーブルは使用可能か)	yes, difficult, no
Frame (Frameは使用可能か)	yes, no
ScrollBar (スクロール可能か)	yes, no
PointingDevice (使用可能か)	yes, difficult, no

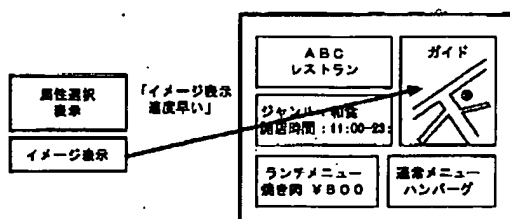
【図21】



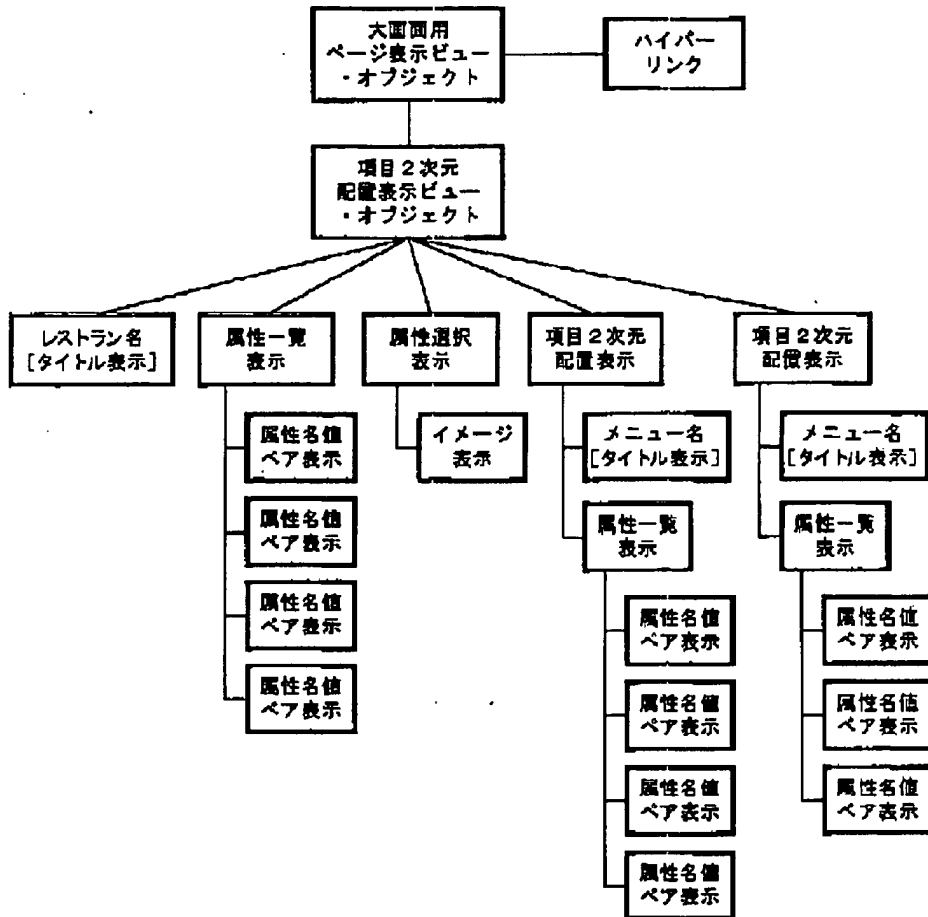
【図22】



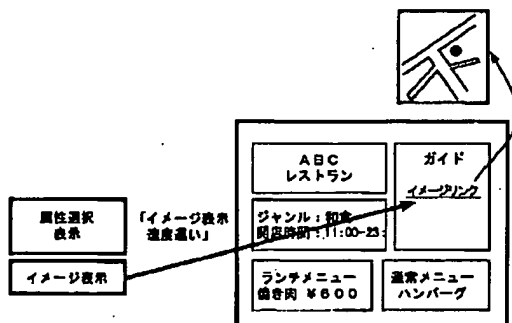
【図25】



【図23】



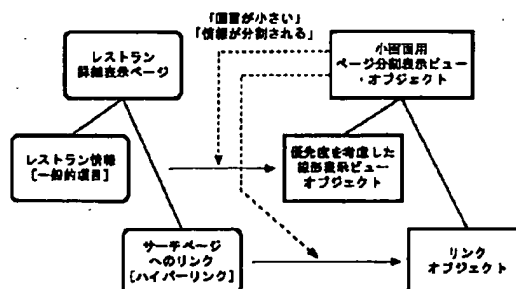
【図26】



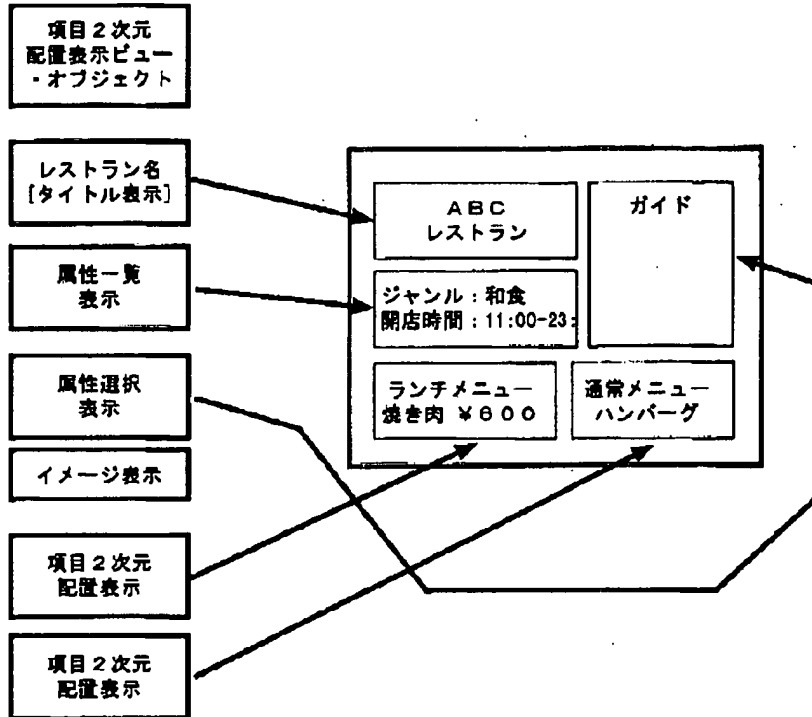
【図27】



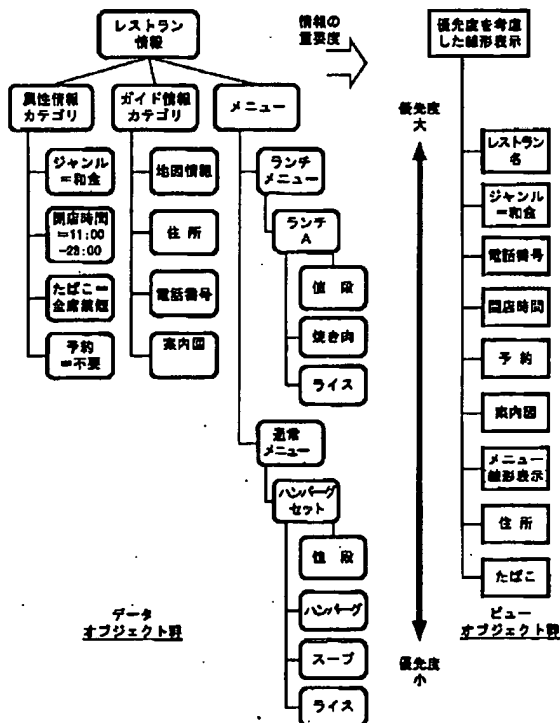
【図28】



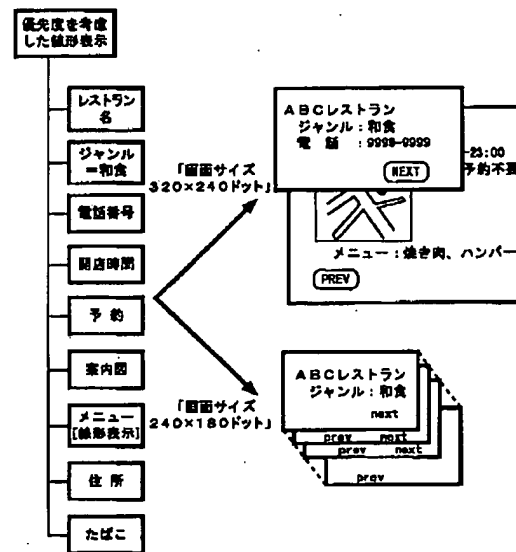
【図24】



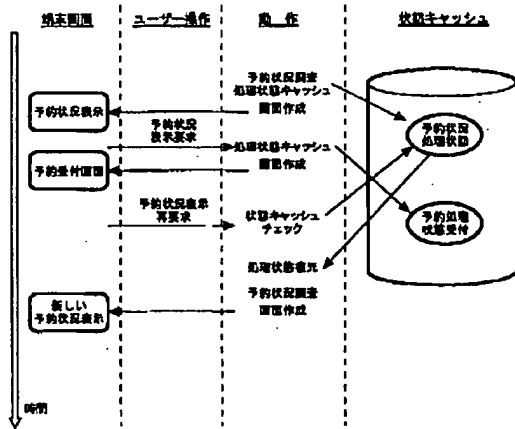
【図29】



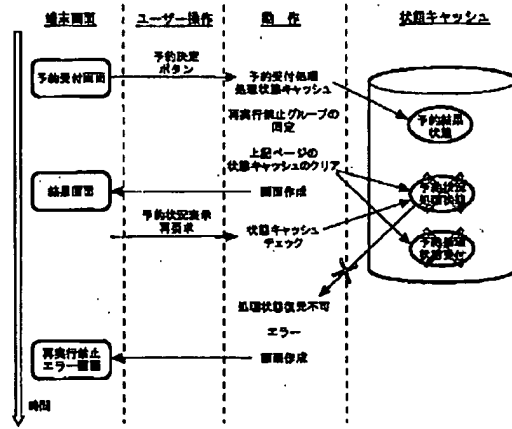
【図30】



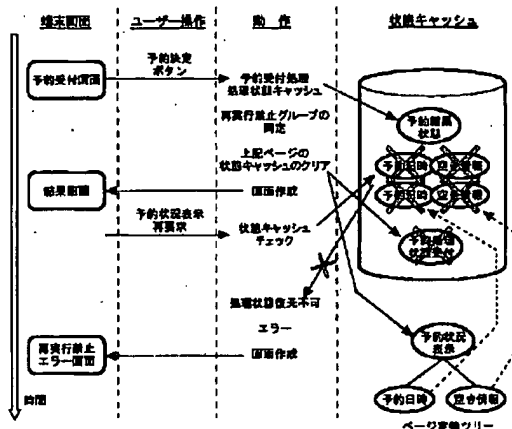
【図31】



【図32】



【図33】



フロントページの続き

(72)発明者 北山 文彦  
 神奈川県大和市下鶴間1623番地14 日本ア  
 イ・ビー・エム株式会社 大和事業所内